Computational Statistics

Potts Clustering with Complete Shrinkage: a novel clustering methodology with its Software Package pottscompleteshrinkage in Python

--Manuscript Draft--

Manuscript Number:	
Full Title:	Potts Clustering with Complete Shrinkage: a novel clustering methodology with its Software Package pottscompleteshrinkage in Python
Article Type:	Original Paper
Keywords:	Potts Model, Complete Shrinkage, bonds, Cluster size, Agglomerative Clustering, Density Curves, Statistical Software, Swendsen-Wang algorithm
Manuscript Classifications:	1.00600: Cluster Analysis; 1.00800: Data Analysis; 1.04090: Shrinkage Methods; 1.04470: Statistical Algorithms; 1.04490: Statistical Computing; 1.04520: Statistical Graphics; 1.04580: Statistical Software; 1.05000: Visualization
Corresponding Author:	Nonvikan Karl-Augustt ALAHASSA University of Montreal: Universite de Montreal Montreal, Québec CANADA
Corresponding Author Secondary Information:	
Corresponding Author's Institution:	University of Montreal: Universite de Montreal
Corresponding Author's Secondary Institution:	
First Author:	Nonvikan Karl-Augustt ALAHASSA
First Author Secondary Information:	
Order of Authors:	Nonvikan Karl-Augustt ALAHASSA
Order of Authors Secondary Information:	
Funding Information:	Institut de Valorisation de données (IVADO) — Dr. Nonvikan Karl-Augustt ALAHASSA
Abstract:	We have modified the Potts Model Swendsen-Wang algorithm to insert some clusters constraints by applying a modified agglomerative clustering approach (Kurita, 1991). We have called the induced Potts Model, \emph{the Potts Clustering with Complete Shrinkage} (PCCS), under the Python package pottscompleteshrinkage deployed on PyPi Index under its current release. In this approach, we deal with the increasing number of small clusters generated in a given partition by merging all small clusters of size $\leq \mathbf{h}$ with their closest cluster in terms of minimal distance respectively, where $\mathbf{h}$ is an integer greater or equal to 2. The algorithm uses a technique in which distances of all pairs of observations are stored. Then the nearest cluster (with size $\geq \mathbf{h}$) is given by the cluster with the closest node in terms of minimal distance to the cluster to be merged using complete linkage. This approach is truly effective as it helps to control the clusters size, and we have found empirical evidence of Chi Squared and Gamma density curves for the constrained cluster size distribution of PCCS, when applied to some datasets taken from the multiple-output benchmark datasets available in the Mulan project website (Tsoumakas et al.,2020). We add a last framework based on Frequency of frequency distribution (FoF) to find the conditional bonds distribution given the clusters size constraints which results in an intractable distribution for large datasets, but its computation framework is a land of rich mathematical developments.

Powered by Editorial Manager® and ProduXion Manager® from Aries Systems Corporation

Manuscript (pdf)
Click here to view linked References

Research Article 1 (2021) 1-101
Submitted x/2021; Published x/xx

Potts Clustering with Complete Shrinkage: a novel clustering methodology with its Software Package *pottscompleteshrinkage* in Python

Nonvikan Karl-Augustt Alahassa, Ph. D Candidate ALAHASSAN@DMS.UMONTREAL.CA
Department of Mathematics and Statistics
University of Montreal
Montreal, Quebec, Canada

Editor: xxx

Abstract

We have modified the Potts Model Swendsen-Wang algorithm to insert some clusters constraints by applying a modified agglomerative clustering approach (Kurita, 1991). We have called the induced Potts Model, *the Potts Clustering with Complete Shrinkage* (PCCS), under the Python package **pottscompleteshrinkage** deployed on PyPi Index under its current release. In this approach, we deal with the increasing number of small clusters generated in a given partition by merging all small clusters of size $\leq$ **h** with their closest cluster in terms of minimal distance respectively, where **h** is an integer greater or equal to 2. The algorithm uses a technique in which distances of all pairs of observations are stored. Then the nearest cluster (with size $\geq$ **h**) is given by the cluster with the closest node in terms of minimal distance to the cluster to be merged using complete linkage. This approach is truly effective as it helps to control the clusters size, and we have found empirical evidence of Chi-Square and Gamma density curves for the constrained cluster size distribution of PCCS, when applied to some datasets taken from the multiple-output benchmark datasets available in the Mulan project website (Tsoumakas et al., 2020). We add a last framework based on Frequency of frequency distribution (FoF) to find the conditional bonds distribution given the clusters size constraints which results in an intractable distribution for large datasets, but its computation framework is a land of rich mathematical developments.

Keywords: Potts Model, Complete Shrinkage, bonds, Cluster size, Agglomerative Clustering, Density Curves, Statistical Software, Swendsen-Wang algorithm.

©2021 alahassan.

Contents

1 Introduction **3**
- 1.1 On Clustering and the random Potts models: . . . 3
- 1.2 Agglomerative Clustering and Shrinkage methods: . . . 4
- 1.3 On some software available for the Potts model: . . . 5

2 The Potts Clustering in its detailed description **6**
- 2.1 The Bernouilli bonds . . . 6

3 Notes on Standard Application: Random Partitions Models **7**

4 Other variants of the Potts Model with their applications **9**

5 Introduction to Kernels **10**

6 Shrinkage algorithms and the Potts Complete Shrinkage model **11**
- 6.1 On Potts Model with Complete Shrinkage Clustering . . . 13

7 Turning some parameters **14**
- 7.1 The optimal number of Potts States . . . 14
- 7.2 Choosing the bandwidth and the temperature . . . 14
- 7.3 Phase transitions . . . 15

8 Code layout, usage and Effective Python Implementation **15**
- 8.1 Code Main layout: modules and methods . . . 15
- 8.2 Code Usage . . . 16

9 Experiments and Examples of Usage of the Package **19**
- 9.1 The plots and description graphics . . . 20

10 Extended Research on the conditional bonds distribution given the components size constraint **27**
- 10.1 Frequency of frequencies distribution . . . 27
- 10.2 Objective . . . 27
- 10.3 Methodology and combinatorial approach to the count vector . . . 28
  - 10.3.1 The FoF distribution for all subsets (of data) of unique label (at initial labelling) . . . 28
  - 10.3.2 FoF distribution for the Potts clusters . . . 29
- 10.4 The conditional bonds distribution (given the size constraint) . . . 30
  - 10.4.1 Computation method . . . 30
- 10.5 Fast-Algorithm to find S_c-list . . . 31
- 10.6 Finding the most probable configuration among S_c-list given a label assignment . . . 32

11 Conclusion and final notes **33**
- 11.1 Notes on the Potts Models and the constrained bonds distribution. . . . 33
- 11.2 The Potts Model with Complete Shrinkage: . . . 33
- 11.3 On further releases of the software package *pottscompleteshrinkage* . . . 34
  - 11.3.1 On the possibility to include other algorithms at the Shrinkage step: . . . 34

12 Acknowledgments **34**

1. Introduction

The Python program **Pottscompleteshrinkage** is designed to constrain the size of each component in the Potts Clustering model known as random partitioning.

1.1 On Clustering and the random Potts models:

Clustering is the method of categorising data into groups or clusters such that objects within a cluster have a high degree of similarity to one another but are quite different from objects in other clusters. The term cluster analysis itself encompasses a number of different algorithms and methods (Tree Clustering [Lin et al. (2018), Liu et al. (2005), Freeman (2006), Ahmed et al. (2011), Lv et al. (2018b), Freeman (2007), WANG et al. (2009), Buttrey & Whitaker (2015), Qiu & Li (2021), Jothi et al. (2015), Page (1974), Vathy-Fogarassy et al. (2005), Miller & Rose (1994)], Block Clustering, k-Means Clustering Wilkin & Huang (2007) and EM algorithms) for grouping objects of similar kind into respective categories, graph-based clustering (Bai et al. (2017)), hierarchical clustering (Köhn & Hubert (2014)), model-based clustering (Fraley & Raftery (1998), Fraley & Raftery (1999), Fraley & Raftery (2002)); Lloyd's *K*-means clustering and the progressive greedy *K*-means clustering (Wilkin & Huang (2007)).

All those clustering techniques and algorithms (Rodriguez et al., 2019) have been applied to a wide variety of research problems (Mann & Kaur (2013), Tan et al. (2013), Fasulo (1999), Jacquez (2008), Blashfield & Aldenderfer (1988), Jacquez (2008), Blashfield & Aldenderfer (1988)); with many books, including (Anderberg (2014), Hartigan (1975), Hennig et al. (2015)) and software (De Hoon et al. (2004)) for applications purposes.

Among those techniques, two particular models on clustering are of interest: the Ising Model and the Potts Model, which are both known as *Restricted Boltzmann machines (RBMs)* (Goel, 2020). The Ising model is very famous as a model named after the physicist Ernst Ising: in 1925, Ernst who was a Lenz student, chose the model as the focus of his PhD dissertation (Kobe (2000), Kobe (1997), Ising (1925), Brush (1967), Ising et al. (2017)). It is a NP-complete (Cipra, 2000) mathematical model of ferromagnetism in statistical mechanics (Singh (2020), Giacomin & Mahfouf, Cipra (1987), Glauber (1963), Pfeuty (1970), Brush (1967), Fredrickson & Andersen (1984), Stauffer et al. (1993), Kadowaki & Nishimori (1998),Joya et al. (2002), Aiyer et al. (1990), Wen et al. (2009), McCoy & Wu (2014)):

$$H = -\left(\sum_{i<j} \Delta_{ij} v_i v_j + \sum_i \delta_i v_i\right) \tag{1}$$

Where Δ_{ij} is to quantify the link, bond or connection between node j and node i, $v_i \in \{0,1\}$, is the state of unit i, $\{0,1\}$ a set of possible[1] states for v_i, δ_i represents a bias from unit i in the system ($-\delta_i$ is the activation threshold for the unit i, an external field). H is the Hamiltonian of the model.

As explained with equation [1], the number of states possible to unit i is limited to two (2) for the Ising Model: $v_i \in \{0,1\}$. The Potts Model is an extended version of this model, in which the number of possible state for unit i is finite and equal to q. The Potts model is a probabilistic system model created by particles which are data points, and a similarity measure controls their interactions. For a set of data $\{x_i\}_{1:N}$, with $x_i \in \mathbf{R}^d, i = 1, \ldots, N$, we denote $\Xi_{ki} = 1$ if x_i belongs to the k-th cluster, and set our Δ_{ij} in equation 1 to be $\Delta_{ij} = \Delta_{ij}(i, j, \Xi_{ki})$. We can see the product $v_i v_j$ as a similarity measure, and replace it by one more general notation $v_i v_j = s\left(x_i, x_j\right)$. The temperature η is a critical parameter for the

1. It may be infinite for a General Boltzmann Machine.

system (Baxter (1973)), and all external fields $-\delta_i$ in 1 are equal to zero. For each η, there is a probability $p_\eta\left(\{\Xi_{ki}\}\right)$ associated with each configuration of the system (Murua et al. (2008a)):

$$p_\eta\left(\{\Xi_{ki}\}\right) \propto \exp\left\{-\frac{1}{\eta}H\left(\{\Xi_{ki}\}\right)\right\} = \exp\left\{-\frac{1}{2\eta}\sum_{i=1}^{n}\sum_{j=1}^{n}\Delta_{ij}(i,j,\Xi_{ki})s\left(x_i,x_j\right)\right\}$$

The Potts model has the property to return less-likely configurations that assign different labels (clusters) to observations that are similar (Ashkin & Teller (1943); Graner & Glazier (1992); Selke & Huse (1983)). Its applications extend to Some of the popular unsupervised learning methods are clustering, dimensionality reduction, image segmentation, association mining, anomaly detection and generative models (Li & Lowengrub (2014), Blatt et al. (1997); Reichardt & Bornholdt (2004); Kullmann et al. (2000); Grimmett (1994); Asikainen et al. (2003); Coniglio & Peruggi (1982); Georgii & Häggström (1996); Fortuin & Kasteleyn (1972); Murua & Wicker (2014a); Machta et al. (1996); Sweeny (1983); Cardy & Ziff (2003); Grimmett (2004); Tomita & Okabe (2001); Blatt et al. (1996b); Blatt et al. (1996a); Janke & Schakel (2004); Duminil-Copin et al. (2017)). Each of these techniques has a different pattern recognition objective such as identifying latent grouping, identifying latent space, finding irregularities in the data, density estimation or generating new samples from the data.

As the temperature affects the energy of the system, one difficulty often encountered is that small clusters are dominant in the disordered phase of the Potts Model (Lima, 2018). To deal will it, we apply a modified agglomerative clustering approach (Kurita, 1991) by merging all small clusters of size $\leq$ **h** with their closest cluster in terms of minimal distance respectively, where **h** is an integer greater or equal to 2. For the merging process, we have applied to the clusters a shrinkage clustering optimization algorithm method **completed linkage** [6], inspired from available agglomerative clustering and shrinkage methods.

1.2 Agglomerative Clustering and Shrinkage methods:

Once the clustering constraint has been fixed, we can rely on agglomarative methods to merge those clusters that do not fit the given constraint, with others that satisfy it.

Agglomerative clustering is a set of methods that generate the partition hierarchically by a sequence of merge operations (Fränti & Virmajoki (2006), Ackermann et al. (2014), Sasirekha & Baby (2013), Madhulatha (2012), Gower (1967), Wattanachon & Lursinsap (2004), Rani[1] & Rohil (2013), Patel et al. (2015)). At the onset, hierarchical agglomerative clustering treats given data as a singleton cluster, then merges (or agglomerates) pairs of clusters until all clusters are merged into a single cluster containing all data (Lukasová (1979), Müllner (2011)).

Hierarchical clustering algorithms can also be used for graphical data exploration (Seo & Shneiderman (2003), Sprenger et al. (2000), Wu et al. (2010)). One of them known as Single-linkage (Ross, 1969), can easily depict the hierarchical relationship of clusters based on a similarity measure. When error are accumulated during the agglomeration process, Jin & Xiao (2013) has developed an adaptive hierarchical agglomerative clustering algorithm called Agglomerative Network Clustering Algorithm (ANCA) from Newman Rapid Algorithm Based on Heap (NRABH) to eliminate the error in advance.

Shrinkage methods are useful when it is time to apply the appropriate merging criteria, **i.e**, to decide whether two clusters are relevant to be merged together. The idea of shrinkage in clustering originates from convex optimization view towards the task of clustering ((Pelckmans et al., 2005)). In terms of clustering, related works are numerous: shrinkage clustering based on matrix factorization that simultaneously finds the optimal number of clusters while partitioning the data (Hu et al., 2018), James–Stein

shrinkage in k-means clustering where the centroids of clusters toward the overall mean of all data is shrunk using a James–Stein-type adjustment (Gao & Hitchcock, 2010). In the work of Pelckmans et al. (2005), a shrinkage term is proposed resulting in sparseness amongst the differences between the centroids. Clusters are obtained by solving a convex optimization problem with a fixed trade-off term between clustering loss and the shrinkage term. A hierarchical clustering tree is then generated by varying the trade-off value.

When it comes to practice, shrinkage techniques are also considered as a major part of regularization methods, with applications in many statistics related fields such as regression, times series, machine learning, multivariate inference and optimization methods: Tibshirani (1996), Irfan et al. (2013), Van Erp et al. (2019), Similä & Tikka (2007), Gruber (2017), Steyerberg et al. (2001), Ahmed & Nicol (2012), Ahmed (1997), QIAN & Su (2016), Sætrom & Omre (2011), Thompson (1968), Sundberg (2006), van Houwelingen & Sauerbrei (2013), Ahmed (2014), Polson & Scott (2012), Zareamoghaddam et al. (2020), Yüzbaşı et al. (2020), Agarwal (2002), Lian (2013), Zheng et al. (2014), Roozbeh & Arashi (2016), Xiong & Joseph (2013), Tutz & Leitenstorfer (2006), Griffin et al. (2017), Korobilis (2013), Korobilis (2013), Jiang & Owen (2003), Zou & Hastie (2003), Efron (1992), Fan et al. (1991).

In our case, we apply shrinkage methods with linkage based-algorithms [6].

1.3 On some software available for the Potts model:

Software are required to be more adequate both for learning and doing statistics in introductory courses. At the most practical level, software should be used to encode model equations and methods (or, more usefully, as mark-up languages for generating software). Those software have due requirements Biehler (1997).

Among the software available for Potts Models, we have those developped in **R** language : **bayesImageS** for Bayesian Methods for Image Segmentation (Moores & Mengersen (2018), Moores et al. (2016)), **potts** for Markov Chain Monte Carlo for Potts Models (Geyer & Johnson, 2010), **PottsUtils** for Utility Functions of the Potts Models (Feng & Tierney (2014), Feng). This package [PottsUtils] introduces also some variants of the Potts model: *simple*, *compound* and *repulsive* Potts models (Feng, 2008).

Pottslab is another one, developed by Storath & Weinmann (2014), and is used as a Matlab/Java toolbox for the reconstruction of jump-sparse signals and images using the Potts model (also known as "piecewise constant Mumford-Shah model" or "l0 gradient model"). Applications include denoising of piecewise constant signals [where L1 Potts model is robust to noisy and moderately blurred data], step detection [Szorkovszky et al. (2018), Nord et al. (2017)] and segmentation of multichannel image (Breger et al., 2017).

Parts of **Pottslab** can be used without Matlab as pure Java plugins, such as Icy plugin - an interactive image segmentation plugin based on **Pottslab** (Implemented by Vasileios Angelopoulos at the Biomedical Imaging Group (BIG), EPFL in Switzerland) and ImageJ plugin - an ImageJ frontend for Pottslab (written by Michael Kaul).

Some others software have been developped for the Cellular Potts model: **CompuCell3D** (Cickovski et al. (2007), Swat et al.), CompuCell (Izaguirre et al., 2004) and the two dimensional Cellular Potts Model Library called Tissue Simulation Toolkit based on the work of Graner and Glazier (Graner & Glazier, 1992).

Our paper works & contributions: The following lines introduce the Potts Clustering in its detailed description [2], describe most of its applications [3, 4], introduce some kernels [5], and shrinkage algo-

rithms [6]. Our model is presented in section [6.1], and its code/package usage is explained in [8]. The model needs the user to turn some key parameters [7], and experiments [9] are given to illustrate the package usage. More developments are proposed on the conditional bonds distribution in [10]. Our final words follow in section [11].

2. The Potts Clustering in its detailed description

The Potts Clustering is a random partition model for clustering with the prior distribution on partitions ρ_n being the Potts Model.

Let $\boldsymbol{\mathcal{D}} = \{x_i \in \mathbf{R}^p, i = 1, ..., n\}$ be our data, i.e, the observations form the vertices of a graph (as in the super-paramagnetic clustering framework). Let us denote this data graph by $(\boldsymbol{\mathcal{G}}(\boldsymbol{\mathcal{D}}), (\boldsymbol{\mathcal{G}}))$, where $(\boldsymbol{\mathcal{G}})$, the edge-set is composed of pairs of nearest neighbors, i.e:

$$(\boldsymbol{\mathcal{G}}) = \{(x, y) \in \boldsymbol{\mathcal{D}}^2, k_{xy}(\sigma) = k_\sigma(x, y) > 0\}$$

where k_{xy} is said to be the similarities between the neighboring points x and y (k_{xy} a Mercer kernel in (x, y) given a bandwidth parameter σ). If x and y are neighboring points, we will write $x \sim y$.

In Potts clustering, we assign labels $i \in \{1, ..., q\}$ to each observation $x_i, i = 1, ..., n$, so that observations similar to each other are likely to be assigned the same label. Denoting $z_{si} = 1$ if x_i has been assigned to the sth label, and zero, otherwise, the model density is given by:

$$p(\{z_{si}\}|\sigma, X, \beta, q) = Z^{-1} \exp\Big\{ -\beta \sum_{x_i \sim x_j} k_\sigma(x_i, x_j)(1 - \delta(x_i, x_j))\Big\}$$

where $\beta = \frac{1}{T}$ is the inverse temperature parameter, $\delta(x_i, x_j) = \sum_{s=1}^{q} z_{si} z_{sj} = 1$ if x_i and x_j have the same label assignment.

2.1 The Bernouilli bonds

Let's introduce *percolation* (Duminil-Copin, 2016).

Definition 2.1 (Percolation configuration)**.** A percolation configuration $b = (b_{ij} : (x_i, x_j) \in (\boldsymbol{\mathcal{G}}))$ is an element of $\{0, 1\}^{(\boldsymbol{\mathcal{G}})}$. If $b_{ij} = 1$, the edge (x_i, x_j) is said to be frozen (open), otherwise (x_i, x_j) is said to be not frozen (closed).

A percolation model is given by a family of probability measures on percolation configurations. Giving The Kastaleyn-Fortuin mapping, which establishes a connection between a particular percolation model and a limit of the Potts model (Hu, 1987), Bernouilli bonds are introduced to match the percolation configuration for the Potts clustering.

For an edge $(x_i, x_j) \in (\boldsymbol{\mathcal{G}})$, the bond b_{ij} becomes frozen with conditional probability (given δ_{ij}):

$$p_{ij} = p(b_i j|\delta_{ij}) = \delta_{ij}\big(1 - \exp\{-\beta k_{ij}(\sigma)\}\big)$$

where $\delta_{ij} = \delta(x_i, x_j)$ and $k_{ij}(\sigma) = k_\sigma(x_i, x_j)$.

This is known as the Fortuin-Kasteleyn-Swendsen-Wang model (Sokal, 1997).

Maximal connected components (clusters) are obtained by finding all observations in a frozen path (Murua & Quintana, 2017b),i.e, each cluster is then identified by those unique observations among a

given frozen path. Isolated observations form a cluster of size equal to 1. This is a major drawback, because in real applications (biomedical datasets (Hu et al., 2018), finance, computer science, engineering (Ganganath et al., 2014)), it is often essential to obtain clusters of sufficient sample size to make the clustering result meaningful and interpretable for subsequent analysis.

Because the bonds probabilities are conditioned by the label assignment process, it is obvious that the clustering is influenced by q, the number of labels. The larger q is, the more subsets (of data) of unique label are generated at random.

There is a need to control even the mean size of all subsets (of data) of unique label, after **initial label assignment**. We only care about the initial labelling because for further steps (in the case of Swenden-Wang algorithm) [Borgs et al. (2012); Salas & Sokal (1997); Häggkvist et al. (2004); Häggström et al. (2002); Johansson & Pistol (2011); Martinelli et al. (1990); Ding & Barbu (2015); Häggkvist et al. (2004)], each connected subset is assigned the same color label uniformly at random and independently from each other, given the bonds $\{b_{ij}\}$.

---

3. Notes on Standard Application: Random Partitions Models

It is well-known that a random measure in Bayesian non-parametrics induces a distribution over random partitions. Many Random partition models do exist with multiple applications (Dahl et al. (2017); Dahl (2008); Loschi & Cruz (2005); Betancourt et al. (2020); Di Benedetto et al. (2017); McCullagh (2011); Zanella et al. (2015); Stam (1983)). Some random partitions are implied by the Dirichlet process (DP) prior $p(\pi_n)$ (Blackwell et al. (1973); Ferguson (1973); Antoniak (1974); Müller & Quintana (2010)). The most famous random partitions model, is the one of Müller & Quintana (2010), which introduced a *cohesion* measure :

$$P(\pi_n = \{S_1, \ldots, S_\Xi\}) = K \prod_{\xi=1}^{\Xi} c\left(S_\xi\right) \tag{2}$$

where π_n is a partition of the objects in a family of subsets $S_1, S_2, \ldots, S_\Xi$ of $S_0 = \{1, 2, \ldots, n\}$ and $c(S)$ is a non-negative cohesion that is specified for each subset of S_0, $\Xi = |\rho|$ is the number of partitions. Here, the normalizing constant $K = \sum_{\rho \in \mathcal{P}} \Pi_{\xi=1}^{\Xi} c\left(S_\xi\right)$, where $\mathcal{P}$ is the set of all possible partitions into non-empty sets.

> As a reminder : Cohesion is the measure of the strength of the functional relationship of the elements in each subset that then controls the partition of subsets that can be roughly thought of as a probability (Page et al. (2019); Müller et al. (2013)).

A popular choice is $c(S) = m(|S| - 1)!$ where m is a precision parameter and $|S|$ is the number of elements in S. It follows that the resulting probability model for π_n is

$$P(\pi_n) = \frac{m^{\Xi - 1} \prod_{\xi=1}^{\Xi} (n_\xi - 1)!}{\prod_{i=1}^{n} (m + i - 1)}$$

where $n_\xi = \left|S_\xi\right|$ is the number of elements in cluster j that is known as the Dirichlet process (DP) random partition.

Remark (Dahl et al. (2009))
The connection between product partition models and Dirichlet process mixture (DPM) models was first shown by Quintana & Iglesias (2003). The proof is obvious. take the equation 2, and replace $c(S) = m(|S| - 1)!$, we get :

$$P\left(\pi_n = \left\{S_1, \ldots, S_{\xi}\right\}\right) = K \prod_{\xi=1}^{\Xi} m(|S_{\xi}| - 1)! = m^{\Xi} \cdot K \prod_{\xi=1}^{\Xi} (|S_{\xi}| - 1)!$$

And it is easy to find the right K, that will make $P\left(\rho_n = \{S_1, \ldots, S_{\Xi}\}\right)$ a probability:

$$K = \prod_{i=1}^{n} (m + i - 1)$$

.

Among many related random partition models, we have :

1. Product partition models (PPM) [Hartigan (1990); Barry & Hartigan (1992); Dahl et al. (2009); Loschi & Cruz (2005); Loschi & Cruz (2002); Quintana & Iglesias (2003)] is a special case. These model assume that observations in different elements of a random partition of the data are independent. So if the probability distribution for the random partitions is in a product form prior to obtaining observations, it is also then in product form after obtaining the observations (Jordan et al. (2007)).

   Definition 3.1 (Product Partition Model)**.** Together with independent sampling across clusters, a ***PPM*** can be described as (Quintana & Iglesias (2003); Blackwell et al. (1973); Pitman (1996); Dahl et al. (2009)):

$$\mathcal{P}\left(\mathbf{y} \mid \pi_n\right) = \{S_1, \ldots, S_{\Xi}\}) \propto \prod_{\xi=1}^{\Xi} c\left(S_{\xi}\right) P\left(\mathbf{y}_s\right)$$

2. Product partition models with a covariate-dependent extension (PPMx) proposed by Müller et al. (2011), Dunson & Park (2008), and Dahl(2008). This PPM version uses covariates approach, with preditors dependent random probability measures. In this application, covariates are available and are used to a *priori* inform the clustering. This leads to random clustering models indexed by covariates, i.e., regression models with the outcome being a partition of the experimental units. There are many PPMx variants : Fung (2012), Quintana (2010); Quintana et al. (2020); Blei & Frazier (2011); Jo et al. (2015); Barcella et al. (2017); Ferreira et al. (2014); Page & Quintana (2018). One standard application of the Potts Model, is in fact to build random clustering covariates model with regression. The main example is the model of Murua & Quintana (2017b). We introduce briefly the scope of their model: suppose we have a set of n data available, and that each individual $i \in [n]$ in a given sample is associated with a p -dimensional vector y_i of responses of interest and a q-dimensional covariate vector x_i. Suppose also that the interest lies in studying the relationship between y_i and x_i, and in particular, in predicting the response y_{n+1} associated with a covariate vector x_{n+1} of a future individual. Let $p\left(y_i \mid x_i, \Phi_i\right)$ be a likelihood model stating the relationship between the i th response and the associated covariate vector. The covariate dependent random partition model with a hierarchical structure for these data is as follow:

$$\begin{aligned} y_1, \ldots, y_n \mid \rho_n, \boldsymbol{\Phi}_1^*, \ldots, \boldsymbol{\Phi}_{k_n}^* &\overset{\text{ind}}{\sim} p\left(\boldsymbol{y}_i \mid \boldsymbol{x}_i, \boldsymbol{\Phi}_{s_i}^*\right) \\ \boldsymbol{\Phi}_1^*, \ldots, \boldsymbol{\Phi}_{k_n}^* &\overset{\text{iid.}}{\sim} p(\boldsymbol{\Phi}) \text{ and } \rho_n \sim p\left(\rho_n \mid \boldsymbol{x}^n\right) \end{aligned}$$

Here ρ_n is a partition of $[n]$ into k_n subsets. Also, $s_1, \ldots, s_n$ are cluster membership indicators such that $s_i = j$ if the i th individual belongs to the j th cluster. In addition $\Phi_i = \boldsymbol{\Phi}^*_{s_i}$ for all $i \in [n]$. Model (1) groups in cluster j those individuals having identical parameter value Φ^*_j. Individuals within this cluster are conditionally iid given Φ^*_j. What make the model particular here is that $\rho_n \sim p(\rho_n \mid \boldsymbol{x}^n)$ is a Potts Model. This implies the existence of auxiliary binary variables, the so-called bonds $\boldsymbol{b} = \{b_{ij}\}$, so that:

$$p(\rho_n \mid x^n) = \sum_{\boldsymbol{b} \Rightarrow \rho_n} p(\boldsymbol{b} \mid \boldsymbol{x}^n)$$

They then apply the Metropolis-Hastings (MH) algorithm to sample parameters from this posterior, by choosing an efficient MH proposal distribution and they obtain consistent improvements compared to the results found in the literature. The model simultaneously allows for explicit estimation of the number of clusters, and for good responses predictions (Murua & Quintana (2017b)).

4. Other variants of the Potts Model with their applications

The Potts model (Wu, 1982) uses the Swenden-Wang Monte Carlo algorithm to simulate random partitions with the data based on the ferromagnetic Potts model on a graph. In the Potts model there are q possible states for each spin and the interaction energy between any pair of neighboring spins is $-J$ if the spins have the same value and is 0 otherwise. The 2-state Potts model is equivalent to the Ising model (Duminil-Copin, 2017). In the Ising model, usually the spins are arranged in a square lattice where they interact with other spins within a defined neighbourhood. However, the Potts model has a much richer phase structure, which makes it an important testing ground for new theories and algorithms in the study of critical phenomena. The scope of research for the q-state Potts Model extend to its critical manifolds (Scullard & Jacobsen, 2016), its topological phases in the antiferromagnetic configuration (Zhao et al., 2018), its static critical behavior in high resolution (Caparica et al., 2015), fraction of uninfected walkers in its one-dimensional model (O'Donoghue & Bray, 2002), its ferromagnetic states with multisite interaction (Schreiber et al., 2018), its disordered states without a ferromagnetic phase (Marinari et al. (1999), Carlucci (1999)), approximate theories of first-order phase transitions on its two-dimensional model (Dasgupta & Pandit, 1987) [which has Critical exponents of domain walls (Dubail et al., 2010), critical polynomials Jacobsen & Scullard (2013), entanglement entropy measurable using wavelet analysis Tomita (2018)], periodic p-adic Gibbs Measures Ahmad et al. (2018), local scale invariance in ageing (Lorenz & Janke, 2007), Potts glass models (Yamaguchi, 2015), percolation models on bowtie lattices (Ding et al., 2012), Roughness exponent in two-dimensional percolation, and clock model (Redinz & Martins, 2001), interfacial adsorption in two-dimensional pure and random-bond Potts models (Fytas et al., 2017), exact valence bond entanglement entropy and probability distribution in the XXX Spin Chain (Jacobsen & Saleur, 2008), lung cancer pathological image analysis using a hidden Potts model (Li et al., 2017), the cellular Potts model (He et al. (2009), Durand & Guesnet (2016), Albert & Schwarz (2014), Albert & Schwarz (2014), Voss-Böhme (2012), R. Noppe et al. (2015), Scianna & Preziosi (2014), Harrison & Vasiev (2008)), the two (2)-Dimensional Wetting transition (Lopes & Mombach, 2017), the random resistor network and its Potts-model formulation (Harris & Lubensky, 1987), Directed Small-World Networks (Aquino et al., 2018), Hybrid Potts-Phase Field with Hydride Modeling Capability in MBM (Hales & Tikare, 2014), adaptive Potts model for networks of neurons (Abdollahnia et al., 2012), Parallel family trees for its transfer matrices (Navarro et al., 2015), cortical network (Naim et al., 2018), approximate ground states of the random-field Potts model from graph cuts (Kumar et al., Kumar et al. (2018)), percolation model (Ding et al., 2010), Multiscale multifractal DCCA and complexity behaviors of return intervals for Potts price model (Wang et al., 2018), solution of the sign

problem in the Potts model at fixed fermion number (Alexandru et al., 2018), Swendsen–Wang multi-cluster spin flip algorithm (Komura & Okabe, 2014), aperiodic modulations of the Potts model and its first-order transitions (Branco & Girardi, 2012), semantic memory retrieval (Kropff & Treves, 2005), Simulations of Grain Growth in Copper Interconnects (Radhakrishnan & Sarma, 2008), three-state Potts model (Kolesik & Suzuki (1995), Ferraz & Lima (2017), Handrey Araujo Ferraz & Sousa Lima (2016), Lv et al. (2018a)), metastability and nucleation in the 2D-Potts ferromagnet (de Berganza, 2009), Potts glass reflection of the decoding threshold for qudit quantum error correcting codes (Jiang et al., 2016), localization protection and symmetry breaking in one-dimensional Potts Chains (Friedman et al., 2017), Ground-state entropy of the Potts antiferromagnet with next-nearest-neighbor spin-spin couplings on strips of the square lattice (Chang & Shrock, 2000), Weighted fractional permutation entropy and fractional sample entropy for nonlinear Potts financial dynamics (Xu & Wang, 2017), nonlinear complexity behaviors of agent-based 3D Potts financial dynamics with random environments (Xing & Wang, 2018), Reconstruction of a real world social network using the Potts model and loopy belief propagation (Bisconti et al., 2015), Finite-size behaviour of generalized susceptibilities in the whole phase plane of the Potts model (Pan et al., 2018), Phase diagram of the triangular-lattice Potts antiferromagnet (Jacobsen et al., 2017), chiral Potts model (Chair, 2014), scaling Potts model (Lencsés & Takács, 2014), Radiative corrections to the quark masses in the ferromagnetic Ising and Potts field theories (Rutkevich, 2017), Bayesian image segmentations by Potts prior and loopy belief propagation (Tanaka et al., 2014), quantum Potts chain (Lajkó & Iglói, 2017), Density of states and Potts zeros (Kim & Creswick, 2001), Dynamic metastability in the two-dimensional Potts ferromagnet (Berganza et al., 2014), generative Hopfield-Potts models for protein families (Shimagaki & Weigt, 2019), and Bayesian Methods for Image Segmentation using a hidden Potts model which can be viewed as a spatially-correlated generalisation of the finite mixture model (Moores et al. (2020)).

5. Introduction to Kernels

A couple of kernels have been developped for clustering tasks for multiple applications (Zhang et al. (2002), Langone et al. (2016), Girolami (2002), Graves & Pedrycz (2007), Camastra & Verri (2005)). In machine learning, the kernel function is used on each data instance to translate the observations into a higher-dimensional space where they can be separated.

1. Kernel in K-Means: The Kernel is implemented to minimize the following error :

$$E(X) = \frac{1}{2n} \sum_{i=1}^{k} \sum_{\mathbf{u} \in \pi_i} \|\mathbf{u} - \mathbf{v}_i\|^2 \tag{3}$$

   where $\mathbf{v}_i = \frac{1}{|\pi_i|} \sum_{\mathbf{u} \in \pi_i} \mathbf{u}$ is mean of the Voronoi set π_i.

2. Mercer Kernels [for the following, u_i and u_j are two points in the data domain space, σ a parameter]:

   (a) Simple :

$$K^{(\mathrm{p})}\left(\mathbf{u}_i, \mathbf{u}_j\right) = \frac{\left\|\mathbf{u}_i - \mathbf{u}_j\right\|^2}{2\sigma^2}$$

   (b) Polynomial :

$$K^{(\mathrm{p})}\left(\mathbf{u}_i, \mathbf{u}_j\right) = \left(1 + \mathbf{u}_i \cdot \mathbf{u}_j\right)^p, \quad p \in \mathbb{N}$$

(c) Gaussian version :

$$K^{(\mathrm{g})}\left(\mathbf{u}_i, \mathbf{u}_j\right) = \exp\left(-\frac{\left\|\mathbf{u}_i - \mathbf{u}_j\right\|^2}{2\sigma^2}\right), \quad \sigma \in \mathbb{R}$$

In the version **1.0.14** of the **pottscompleteshrinkage** package, only the simple Mercer is implemented.

3. Non linear distance kernel :

$$\left\|\Phi\left(\mathbf{u}_i\right) - \Phi\left(\mathbf{u}_j\right)\right\|^2$$

with Φ a non linear map function.

The list above of kernels does not contend to be complete. Moreover, in any task, the researcher is ask to question the optimality, robustness, and quality of his choice (Chennuru Vankadara & Ghoshdastidar (2019), Yan & Sarkar (2016), Wang et al. (2017)).

6. Shrinkage algorithms and the Potts Complete Shrinkage model

Let introduce a notation (Ackerman et al., 2010) : we considerate a k-clustering $C = \{C_1, C_2, \ldots, C_k\}$ of a data set X, which is a partition of X into k non-empty disjoint subsets of X (so, $\cup_i C_i = X$) . For $x, y \in X$ and clustering C of X, we write $x \sim_C y$ if x and y belong to the same cluster in C and $x \not\sim_C y$, otherwise.

Definition 6.1 (Ackerman et al. (2010))**.** A linkage is the distance between two clusters. It can be expressed as a function as follows:

$$\ell : \{(X_1, X_2, d) \mid d \text{ is a distance function over } X_1 \cup X_2\} \to \mathcal{R}^+ \text{ such that:}$$

1. ℓ is representation independent: For all (X_1, X_2) and (X_1', X_2'), if $(X_1, X_2, d) \cong (X_1', X_2', d')$ (i.e., they are clustering-isomorphic), then $\ell(X_1, X_2, d) = \ell(X_1', X_2', d')$.

2. ℓ is monotonic: For all (X_1, X_2, d) if d' is a distance function over $X_1 \cup X_2$ such that for all $x \sim_{\{X_1, X_2\}} y$, $d(x, y) = d'(x, y)$ and for all $x \not\sim_{\{X_1, X_2\}} y$, $d(x, y) \leq d'(x, y)$ then $\ell(X_1, X_2, d') \geq \ell(X_1, X_2, d)$.

3. Any pair of clusters can be made arbitrarily distant: For any pair of data sets $(X_1, d_1), (X_2, d_2)$, and any r in the range of ℓ, there exists a distance function d that extends d_1 and d_2 such that $\ell(X_1, X_2, d) > r$.

The objects that we consider are pairs (X, d), where X is some finite domain set and d is a distance function d over X. These are the inputs for clustering functions.

Definition 6.2 (Madhulatha (2012))**.** Shrinkage clustering is a size-constrained clustering algorithm

In the Potts Model, we were confronted to size constraint clustering problem, because many small clusters arise from the Potts clustering Swenden-Wang method. To solve this issue, several Shrinkage

Clustering algorithms with base linkage methods are available. Suppose Cluster r is formed from clusters p and q, with n_r being the number of objects in cluster r; u_{ri} is the i-th object in cluster r. The following notation describes the linkages used by the various methods:

1. **Single linkage**, also called nearest neighbor, uses the smallest distance between objects in the two clusters (Ross (1985), Ackerman et al. (2010), Gower & Ross (1969), Massaro (2005), Mohbey & Thakur (2013), Ross (1969),).

$$d(r,s) = \min\left(\text{dist}\left(u_{ri}, u_{sj}\right)\right), i \in (i, \ldots, n_r), j \in (1, \ldots, n_s)$$

2. **Complete linkage**, also called farthest neighbor, uses the largest distance between objects in the two clusters (Krznaric & Levcopoulos (1998), Dawyndt et al. (2005), Grosswendt & Roeglin (2017), Glasbey (1987), Krznaric & Levcopoulos (1997)).

$$d(r,s) = \max\left(\text{dist}\left(u_{ri}, u_{sj}\right)\right), i \in (1, \ldots, n_r), j \in (1, \ldots, n_s)$$

   In general, the time complexity of complete linkage for N data is at most $\text{O}\left(\text{N}^2 \log \text{N}\right)$ (Day & Edelsbrunner, 1984).

3. **Average linkage** uses the average distance between all pairs of objects in any two clusters (Guenoche (1994), Jarman (2020), Xu et al. (2021), Emmendorfer & de Paula Canuto (2021), Emmendorfer & de Paula Canuto (2021)).

$$d(r,s) = \frac{1}{n_r n_s} \sum_{i=1}^{n_r} \sum_{j=1}^{n_s} \text{dist}\left(u_{ri}, u_{sj}\right)$$

   This method can be compared to the single linkage method (Seifoddini (1988), Seifoddini (1989)).

4. **Centroid linkage** uses the Euclidean distance between the centroids of the two clusters (Jarman (2020), Fung (2001), Mythili & Madhiya (2014), Kellom & Raymond (2017), Dixit & Naskar (2019)).

   $d(r,s) = \|\bar{u}_r - \bar{u}_s\|_2$ where :

$$\bar{u}_r = \frac{1}{n_r} \sum_{i=1}^{n_r} u_{ri}$$

5. Median linkage: to overcome the disadvantage of centroid method the median of two groups are clustered is called median linkage clustering. It uses the Euclidean distance between weighted centroids of the two clusters (Andes (1998)).

$$d(r,s) = \|\tilde{u}_r - \tilde{u}_s\|_2$$

   where $\tilde{u}_r$ and $\tilde{u}_s$ are weighted centroids for the clusters r and s. If cluster r was created by combining clusters p and q, $\tilde{u}_r$ is defined recursively as:

$$\tilde{u}_r = \frac{1}{2}\left(\tilde{u}_p + \tilde{u}_q\right)$$

6. **Ward's linkage** uses the incremental sum of squares, that is, the increase in the total within-cluster sum of squares as a result of joining two clusters (Nielsen (2016), AZIZI et al. (2019), Miyamoto et al. (2015), Szekely et al. (2005), Strauss & von Maltitz (2017), Murtagh (1983), Kimes et al. (2017), Jain et al. (2004), Blashfield (1980), Xu & Wunsch (2008), Murtagh & Legendre (2014)).

   The within-cluster sum of squares is defined as the sum of the squares of the distances between all objects in the cluster and the centroid of the cluster. The sum of squares metric is equivalent to the following distance metric $d(r,s)$, which is the formula linkage uses.

$$d(r,s) = \sqrt{\frac{2n_r n_s}{(n_r + n_s)}} \left\| \bar{u}_r - \bar{u}_s \right\|_2$$

   - where $\| \cdot \|_2$ is the Euclidean distance.
   - $\bar{u}_r$ and $\bar{u}_s$ are the centroids of clusters r and s.
   - n_r and n_s are the number of elements in clusters r and s

   In some references, Ward's linkage does not use the factor of 2 multiplying $n_r n_s$. The linkage function uses this factor so that the distance between two singleton clusters is the same as the Euclidean distance. Ward's linkage clustering often provides strong clustering results in practice (Kimes et al. (2017)).

7. **Weighted average linkage** uses a recursive definition for the distance between two clusters. If cluster r was created by combining clusters p and q, the distance between r and another cluster s is defined as the average of the distance between p and s and the distance between q and s.

$$d(r,s) = \frac{(d(p,s) + d(q,s))}{2}$$

Some shrinkage available software : In addition to the detection of hierarchical structure in complex networks, **SHRINK** has been built by combining the advantages of density-based clustering and modularity optimization methods, in order to identify hubs and outliers (Huang et al., 2010). There is also **SCAN** which solves the same problem, and has been augmented with **gSkeletonClu**, also known as *graph-skeleton based clustering* (Sun et al. (2010), Huang et al. (2012)), which in turn transforms a network clustering problem into the task of finding core-connected components in its associated Core-Connected Maximal Spanning Tree (CCMST).

6.1 On Potts Model with Complete Shrinkage Clustering

One of the difficulties encountered when sampling partitions with the Swenden-Wang algorithm Swendsen & Wang (1987) for the random bond Potts Models as done by Murua & Quintana (2017a), is the distribution of the temperature T of the Potts Model in its probabilistic form as a system of particles (data points) and their interactions given by the similarity measure. The distribution of the system depends on the temperature T. For each T there is a probability $p_T\left(\{z_{kl}\}\right)$ associated with each configuration of the system's labels:

$$p_T\left(\{z_{ki}\}\right) \propto \exp\left\{-\frac{1}{T} H\left(\{z_{ki}\}\right)\right\} = \exp\left\{-\frac{1}{2T} \sum_{i=1}^{n} \sum_{j=1}^{n} \left(1 - \delta_{ij}\right) s\left(x_i, x_j\right)\right\}$$

where $\delta_{ij} = \sum z_{kl} z_{kj}$ between observations i and j equals one if they are assigned to the same cluster k, and zero otherwise. $s\left(x_i, x_j\right)$ is the similarity measure, and finally $z_{ki} = 1$ if observation i belongs to cluster k. Then, as proposed by Murua & Quintana (2017a) (by introducing a set of latent variables, the bonds $\boldsymbol{b}$), the bond $b_{ij} = 1$ is said to be frozen if $b_{ij} = 1$ and $\alpha_{ij} = \delta_{ij} = 1$, that is, the points x_i and x_j are neighbors and have the same label. Otherwise, the bond is not frozen. The bond b_{ij} becomes frozen with probability $p_{ij} = 1 - \exp\left\{-\beta\kappa_{ij}(\sigma)\right\}$.

In our case, the similarities between pairs of covariate vectors are defined by $\kappa_{ij} = K\left(\boldsymbol{x}_i, \boldsymbol{x}_j\right)$ ($K(\cdot,\cdot)$ is a Mercer Kernel). As usual, we have assumed that K is a function of the distances $\left\|\boldsymbol{x}_i - \boldsymbol{x}_j\right\|$, of the form $\kappa_{ij} = \kappa_{ij}(\sigma) = K\left(\left\|x_i - x_j\right\| / \sigma\right)$, where $\sigma > 0$ is a bandwidth parameter.

As the temperature affects the energy of the system, one difficulty often encountered is that small clusters are dominant in the disordered phase of the Potts Model (Lima, 2018). Particularly, we have found in practice that the drawback of the bonds approach is the increasing number of small clusters generated in a given partition. To deal will it, we apply a modified agglomerative clustering approach (Kurita, 1991) by merging all small clusters of size $\leq$ **h** with their closest cluster in terms of minimal distance respectively, where **h** is an integer greater or equal to 2. The algorithm uses a technique in which distances of all pairs of observations are stored. Then the nearest cluster (with size $\geq$ **h**) is given by the cluster with the closest node in terms of minimal distance to the cluster to be merged. We have implemented a **complete linkage** merging process [6], and this is what we call under the clustering model **Potts Clustering with Complete Shrinkage** (PCCS). In our case, the complete shrinkage process is iterative as described in (Fränti & Virmajoki, 2006): iterative shrinking (IS) starts by assigning each data vector to its own cluster. It then removes one cluster at a time until the desired constraint for each cluster has been reached.

7. Turning some parameters

We are about to present some methods to choose some key parameters, as those methods will be implemented in further releases of the software package (pottscompleteshrinkage) of the **Potts Clustering with Complete Shrinkage** model.

7.1 The optimal number of Potts States

Most of the time, the number of Potts states is choosen empirically. As for example, Ekeberg et al. (2013) has applied 21-state Potts models to significantly outperform existing approaches to *direct-coupling analysis*, when treating evolutionarily proteins structures. Gould & Tobochnik (2010) has proposed a Program called **PottsNumberofstates** that uses the Wang-Landau Monte Carlo algorithm to estimate the number of microstates $\Omega(E)$ for each value of E in the q-state 2D Potts model on a square lattice. Such program can be used in practice. Our model is only dedicated to the finite-state Potts Model. However, the number of states can be very large sometimes, and there are inspirational work in this direction when it happens: The Kosterlitz-Thouless transition in two-dimensional abelian spin systems and the Coulomb gas from Fröhlich & Spencer (1981), who dictates that ***the q-states Potts becomes the XY model in large q state*** [4], Microcanonical Entropy of the Infinite-State Potts Model (Johansson & Pistol, 2011), Density of states, Potts zeros, and Fisher zeros of the Q-state Potts model for continuous Q (Kim & Creswick, 2001).

7.2 Choosing the bandwidth and the temperature

Most of the experiments with Potts Model use the simple Mercer Kernel or the Gaussian kernel $J_{ij}(\sigma) = \exp(-\frac{1}{2\sigma^2}\|x_i - x_j\|^2)$, which is the most popular kernel choice for the Potts or super-paramagnetic clus-

tering model (Blatt et al., 1996a,c; Murua et al., 2008b). Although the scale parameter σ may be estimated through a Bayesian stochastic procedure (Murua & Wicker, 2014b), we prefer to use its common estimator which is given by the average distances $\hat{\sigma}^2 = \sum_{i<j} \|x_i - x_j\|^2 / \binom{n}{2}$. Murua & Wicker (2014b) show that the optimal scale is close to this simple estimator.

The temperature of the system is one of the critical parameter of the model (Kinzel & Domany, 1981). Various works have tried to estimate the appropriate temperature: (Zhao & Fu, 2019). However, if $\beta = 1/T$ is treated as unknown, for example in studies using the Bayesian methods (Green & Richardson (2002); Smith & Smith (2006)), then the normalizing constant is necessary when drawing samples from the conditional distribution of β by a Metropolis-Hastings algorithm. The package **PottsUtils** provide a function **getNC()** to obtain the normalizing constant of a *simple* Potts model, using the Bayesian method described similarly in (Green & Richardson (2002); Smith & Smith (2006)).

One of the main issues encountered with the Potts model is the choice of the inverse temperature parameter β. This parameter controls the cluster sizes in the partitions. A value too low of β produces too many small clusters, while a value too high produces very few and large clusters. For example, in a shallow Gibbs network (ALAHASSA & Murua, 2021) which is going to be fitted for each cluster of the partition, we would rather not have too small size clusters. Therefore, an optimal value of β is preferred. In our experiments, for each dataset, we selected a value of β that produced large enough clusters. In general, Murua & Wicker (2014b) gave a simple procedure to find a nearly optimal value of β. In our case, we just chose a value slightly smaller than the value suggested so as to ensure large cluster sizes.

7.3 Phase transitions

We are not interested in our case in the phase transition of the model; much more researches are available on those topics, that the lecturer shall read in the work of : Tan et al. (2020), Giataganas et al. (2021), Ferrenberg & Swendsen (1988), Li et al. (2018), (Gandica & Chiacchiera, 2016), Gorbenko et al. (2018), Gorbenko et al. (2018), Kaufman & Diep (2010), Katori (1988).

The partition function of the Potts Model: The partition function is intractable, but recent advanced research has improved his computation with new algorithms (Hartmann, 2005).

8. Code layout, usage and Effective Python Implementation

The fully implemented architecture and Valid Python Code of the Potts Clustering with Complete Shrinkage is available with comments on our github repository under GNU General Public license ***v*3.0**. Please download the package on the public access link : `https://github.com/kgalahassa/pottscompleteshrinkage` or write directly to alahassan@dms.umontreal.ca for more details.

The full descriptive code of the Potts Clustering with Complete Shrinkage (PCCS) requires at least 100 to 200 lines to present the complete algorithm; the reason why we prefer to (simply) distribute directly the code through github.

8.1 Code Main layout: modules and methods

As presented in figure [1], the Github repository **pottscompleteshrinkage** contains many folders. The main folder which is principal to build the package is the **pottsshrinkage** folder, containing the **__init__.py** file, indicating the **completeshrinkage.py** main module. The other folders such as **Build, dist**, and **pottscompleteshrinkage.egg-info** are built-in folders obtained from two main command lines of ex-

ecution in an Anaconda Terminal: **python setup.py install**, and **python setup.py sdist bdist_wheel**. The main module **completeshrinkage.py** contains all the methods called at usage.

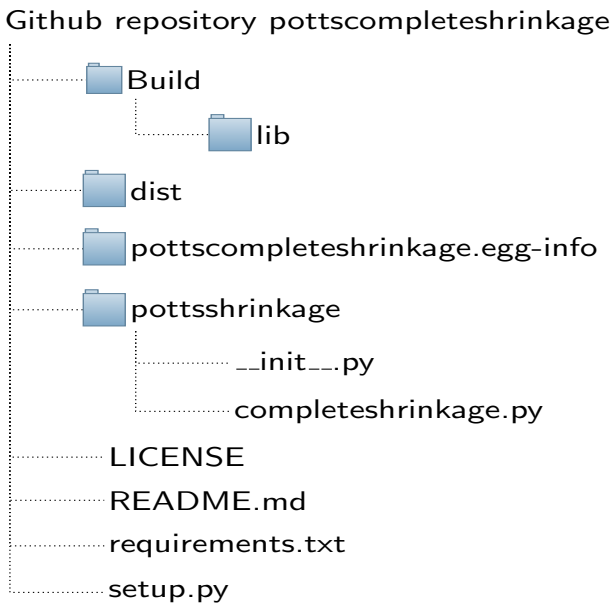


Figure 1: The Potts Complete Shrinkage Github Repository Structure

8.2 Code Usage

The package **pottscompleteshrinkage** is currently published on **PyPi** (The Python Package Index), in its **1.0.14** version (new releases may be published in the future). There is a notebook available at : https://github.com/kgalahassa/pottscompleteshrinkage-notebook to illustrate the usage of the package. There are three requirements before running the code : $q \in \mathbb{N}*$ (q must be a strictly positive integer, $q \geq 10$ is preferable), the temperature T and the bandwidth σ must also be strictly positive real number ($T, \sigma \in \mathbb{R}_+^*$). This notebook is presented and commented in detailed here:

```
import pandas as pd
import random
import numpy as np
from numpy import linalg as LA
from sklearn import datasets

# import some data to play with
iris = datasets.load_iris()
Train_PottsData_demo = iris.data[:, :3]


#Import the Potts Complete Shrinkage module
import pottsshrinkage.completeshrinkage as PCS

#Choose the number of colors
q = 20


#Compute Initial Potts Clusters as a first Random Partition
#(with Potts Model)
InitialPottsClusters =  PCS.InitialPottsConfiguration(Train_PottsData_demo,

q, Kernel="Mercer")

#Choose your temperature (T) level
T = 1000

#Set the bandwidth of the model
sigma = 1

#Set the Number of Random_Partitions you want to simulate
Number_of_Random_Partitions = 3

#Set your initial (random) Potts partition as computed above
Initial_Partition = InitialPottsClusters

#Set the Minimum Size desired for each partition generated
MinClusterSize = 5

#Run your Potts Complete Shrinkage Model to simulate the Randomly

#Shrunk Potts Partitions. Partitions_Sets is a dictionary that

#can be saved with pickle package.

Partitions_Sets,Spin_Configuration_Sets =

PCS.Potts_Random_Partition (Train_PottsData_demo,  T, sigma,
```

```
Number_of_Random_Partitions, MinClusterSize, Initial_Partition,
Kernel="Mercer")
```

```
We are at step: 1
Clusters Size of Current Partition [11, 10, 13, 7, 6, 7, 8, 11, 10, 6, 8, 6, 8,
6, 10, 7, 10, 6]
Partition is: [[34, 9, 12, 1, 29, 3, 42, 13, 38, 8, 41], [39, 7, 23, 37, 4, 35,
47, 2, 6, 22], [40, 17, 0, 28, 27, 36, 31, 20, 46, 21, 19, 32, 16], [48, 10, 5,
18, 33, 14, 15], [49, 30, 25, 43, 11, 24], [89, 53, 80, 59, 64, 26, 104], [95,
61, 82, 71, 88, 79, 44, 116], [97, 75, 58, 54, 74, 65, 127, 91, 56, 51, 85],
[106, 98, 93, 57, 60, 99, 67, 81, 45, 144], [113, 94, 69, 90, 92, 62], [135,
130, 107, 122, 105, 118, 131, 117], [138, 70, 96, 66, 55, 84], [140, 112, 132,
108, 137, 128, 103, 134], [141, 77, 76, 86, 52, 50], [142, 101, 83, 114, 123,
119, 146, 72, 87, 68], [143, 124, 120, 129, 102, 125, 109], [147, 115, 110, 139,
145, 133, 111, 148, 136, 100], [149, 121, 126, 73, 63, 78]]
We are at step: 2
Clusters Size of Current Partition [7, 10, 9, 6, 7, 8, 10, 7, 11, 7, 9, 8, 8, 7,
11, 10, 8, 7]
Partition is: [[39, 7, 23, 36, 28, 27, 20], [40, 17, 0, 35, 37, 4, 42, 6, 2,
22], [45, 12, 1, 29, 3, 38, 8, 13, 41], [46, 21, 19, 32, 16, 44], [48, 10, 5,
18, 33, 14, 15], [49, 30, 9, 25, 43, 26, 11, 24], [89, 53, 69, 80, 59, 81, 60,
79, 34, 76], [95, 61, 94, 88, 93, 47, 58], [99, 67, 82, 71, 92, 62, 98, 57, 64,
31, 86], [121, 96, 66, 55, 90, 84, 106], [126, 73, 63, 78, 97, 75, 54, 74, 65],
[135, 130, 107, 122, 105, 118, 131, 117], [137, 116, 128, 103, 133, 111, 134,
149], [138, 70, 127, 91, 56, 51, 85], [142, 101, 83, 114, 113, 123, 119, 146,
72, 87, 68], [144, 124, 120, 139, 129, 102, 140, 112, 132, 108], [147, 115, 110,
145, 77, 141, 52, 50], [148, 136, 104, 100, 143, 125, 109]]
We are at step: 3
Clusters Size of Current Partition [10, 11, 8, 6, 12, 6, 8, 10, 7, 9, 6, 8, 7,
7, 11, 9, 6, 9]
Partition is: [[43, 26, 11, 24, 40, 17, 0, 37, 4, 44], [45, 12, 1, 30, 29, 3,
42, 13, 38, 8, 41], [46, 21, 19, 32, 36, 31, 20, 18], [48, 10, 16, 33, 14, 15],
[49, 28, 35, 39, 23, 34, 9, 25, 47, 2, 6, 22], [86, 65, 75, 58, 54, 74], [93,
57, 60, 80, 59, 79, 7, 52], [94, 89, 53, 69, 92, 62, 98, 81, 27, 50], [95, 61,
88, 82, 64, 5, 148], [99, 67, 96, 66, 55, 90, 84, 121, 106], [126, 73, 63, 78,
97, 71], [135, 130, 107, 122, 105, 118, 131, 117], [138, 70, 127, 91, 56, 51,
85], [140, 112, 132, 108, 139, 129, 102], [142, 101, 83, 114, 113, 123, 119,
146, 72, 87, 68], [144, 124, 120, 136, 104, 100, 143, 125, 109], [145, 115, 110,
141, 77, 76], [147, 111, 137, 128, 103, 116, 149, 133, 134]]
We are at step: 4
Clusters Size of Current Partition [10, 8, 11, 6, 6, 6, 11, 6, 6, 10, 7, 12, 8,
11, 9, 8, 15]
Partition is: [[39, 7, 23, 34, 9, 25, 43, 26, 11, 24], [40, 17, 0, 35, 36, 31,
28, 27], [45, 12, 1, 30, 29, 3, 42, 13, 38, 8, 41], [46, 21, 19, 32, 16, 44],
[47, 2, 6, 22, 37, 4], [48, 10, 5, 33, 14, 15], [89, 53, 69, 80, 59, 81, 60, 57,
93, 49, 133], [95, 61, 88, 71, 20, 125], [97, 75, 58, 54, 74, 65], [99, 67, 82,
```

```
79, 64, 92, 62, 98, 18, 96], [121, 94, 90, 84, 66, 55, 106], [123, 72, 83, 146,
119, 87, 68, 149, 114, 142, 101, 113], [135, 107, 118, 131, 122, 105, 117, 130],
[138, 70, 126, 73, 63, 78, 127, 91, 56, 51, 85], [141, 77, 76, 86, 52, 50, 145,
115, 110], [144, 124, 120, 139, 129, 102, 143, 109], [147, 111, 137, 116, 128,
103, 134, 140, 112, 132, 108, 148, 136, 104, 100]]
```

The output returns the index of each data observation in each associate cluster, for each generated partition. **Partitions_Sets** is a Python dictionary that can be saved with **pickle** python package.

Spin_Configuration_Sets returns the last spin configuration. In the print above, we can also read the size of each cluster in each obtained partition. It is also easy to use the **time** package to time the process.

9. Experiments and Examples of Usage of the Package

We have performed experiments with ten (9) datasets taken from the multiple-output benchmark datasets available in the Mulan project website (Tsoumakas et al., 2020). The datasets are shown in Table 1. We set the parameters as follows: $q = 20$, $T = 1000$, and $\sigma = 1$.

Table 1: Summary of the ten (10) datasets taken from the Mulan project.

Dataset	Domain	Number of Instances	Number of features	Response variable dimension (targets)
Andromeda	Water	49	30	6
Slump	Concrete	103	7	3
EDM	Machining	154	16	2
ATP7D	Forecast	296	211	6
ATP1D	Forecast	337	411	6
Online sales	Forecast	639	401	12
ENB	Buildings	768	8	2
Water quality	Biology	1 060	14	16
SCPF	Forecast	1 137	23	3

With those data described above, we have generated a set of Potts partitions in a number b, with $b = 50, 100$, with a shrinkage constraint = 3 (a ***minimum*** of two (3) observations per cluster in each partition is required). We focus attention on four (4) characteristics that can help to compare the performance of the algorithm from one given dataset to one another:

- The number of data per cluster (NumbDC): this corresponds to the list of cluster size per partition;
- The Mean Number of data per cluster and per 10 individuals (MeanNumbC10): we take the mean of the list of the number of data per cluster, and divide by 10 [shown in table 2].
- Number of cluster per partition (NumbCPar): this corresponds to a list of the number of clusters per partition taken through all the partitions;
- The Mean Number of cluster per partition (MeanNumbCPar) [shown in table 2];

And finally a histogram to illustrate:

- The number of data per cluster (NumbDC);
- The number of cluster per partition (NumbCPar);

As for example, some histograms have Chi-Square (p) density, with $p = 2$ degree of liberty; or Gamma$(r, 1)$ density, with $r = 1$. Those density curves are identified for andro number of data per cluster [$b = 50$, shrinkage $= 3$] (figure 2) for Water Quality number of data per cluster [$b = 50$, shrinkage $= 3$](figure 18).

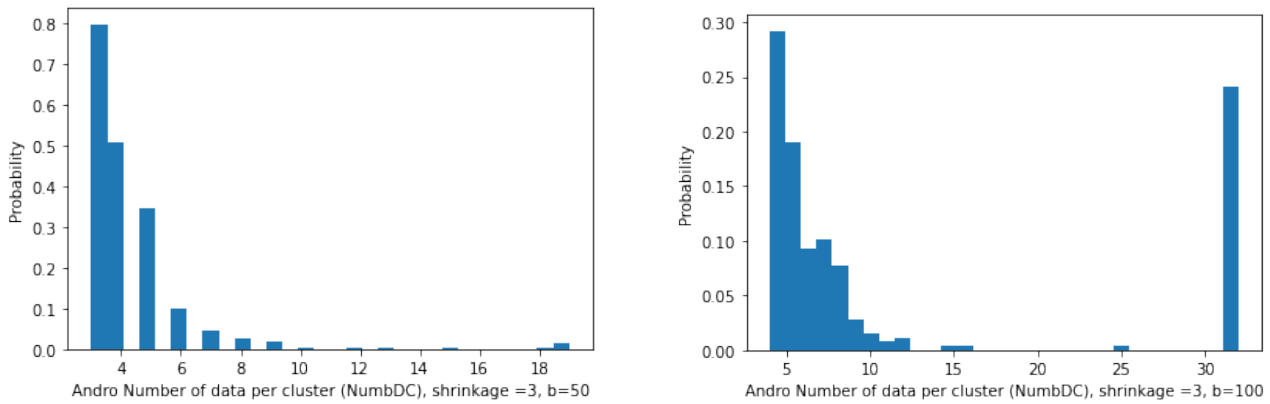
Table 2: Summary of clustering statistics for the Mulan Project Dataset.

	$b = 50$		$b = 100$	
Dataset	MeanNumbC10	MeanNumbCPar	MeanNumbC10	MeanNumbCPar
Andromeda	0.642	4.98	1.175	4.980
Slump	3.007	2.294	3.767	1.831
EDM	0.540	19.058	0.541	19.049
ATP7D*	7.317	2.705	10.049	1.970
ATP1D*	8.964	2.509	8.174	2.509
Online sales*	26.948	1.588	19.829	2.158
ENB	16.691	3.078	27.910	1.841
Water quality	0.536	132.49	0.535	132.702
SCPF	2.647	3.588	3.379	2.812

**We have selected the first six (6) components that explained 80% of the data before the partitioning*

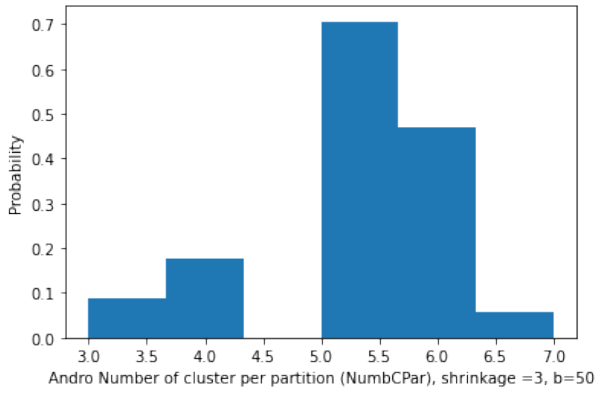
9.1 The plots and description graphics

We have noticed the temperature T and the bandwidth σ are very essential. For higher values of those parameters (the temperature mainly), we get more clusters inside each partition. The configuration ($T \longrightarrow +\infty$ and $\sigma \longrightarrow +\infty$) give more **single** clusters, with the simple Mercer Kernel, and the algorithm takes much more times to apply the shrinkage constraint.

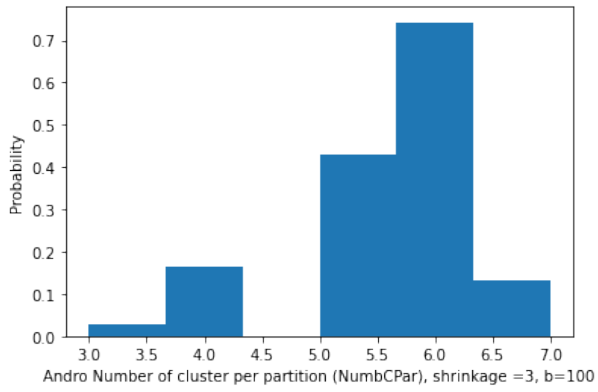

Probability
Andro Number of data per cluster (NumbDC), shrinkage =3, b=50
Probability
Andro Number of data per cluster (NumbDC), shrinkage =3, b=100

(a) $b = 50$ (b) $b = 100$

Figure 2: Andromeda, Number of data per cluster (NumbDC), shrinkage $= 3$

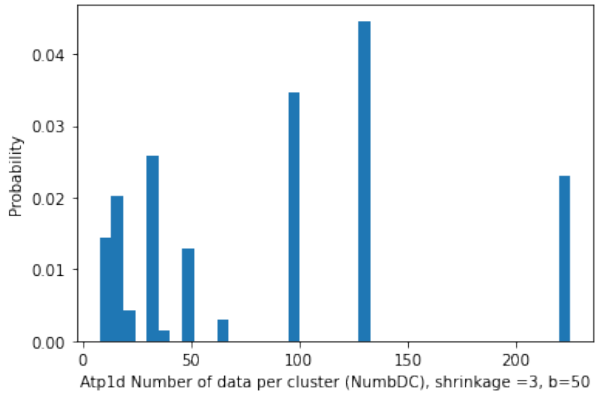


(a) $b = 50$

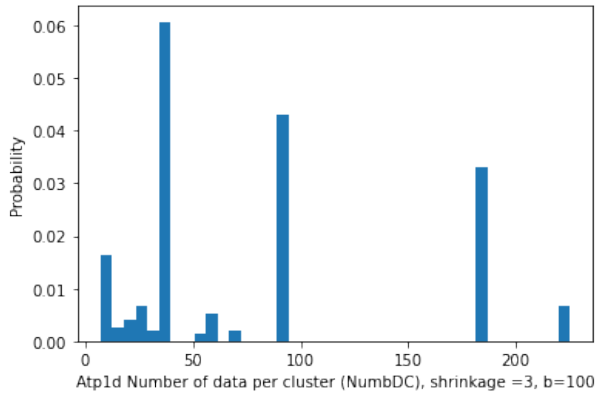


(b) $b = 100$

Figure 3: Andromeda, Number of cluster per partition (NumbCPar), shrinkage $= 3$

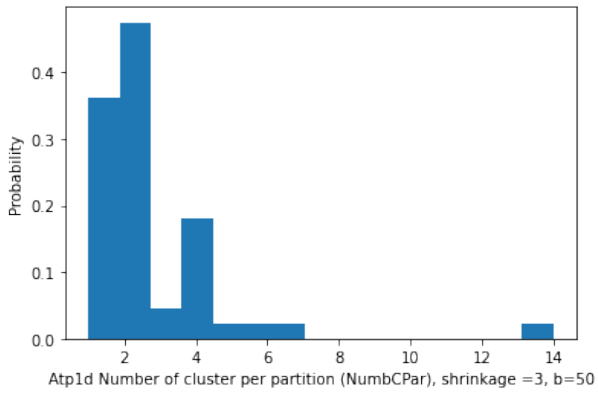


(a) $b = 50$

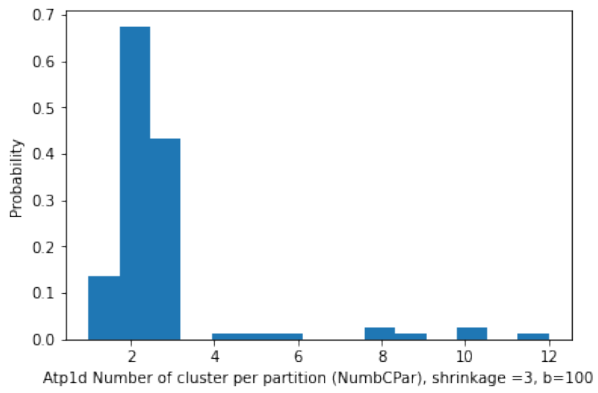


(b) $b = 100$

Figure 4: Atp1d, Number of data per cluster (NumbDC), shrinkage $= 3$

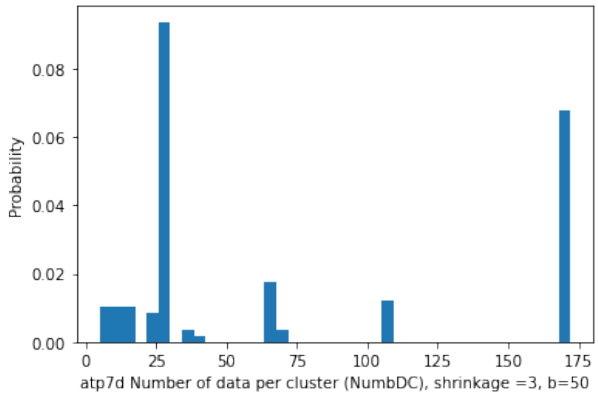


(a) $b = 50$

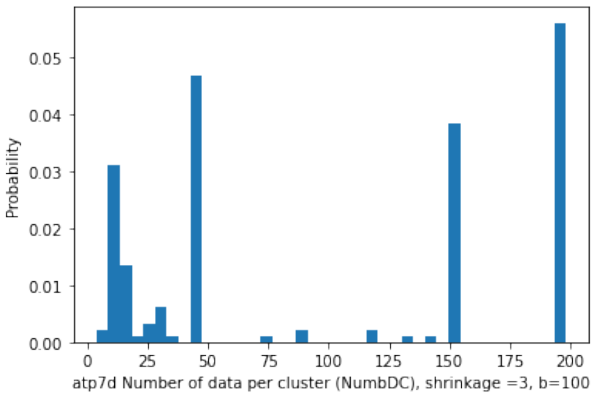


(b) $b = 100$

Figure 5: Atp1d, Number of cluster per partition (NumbCPar), shrinkage $= 3$

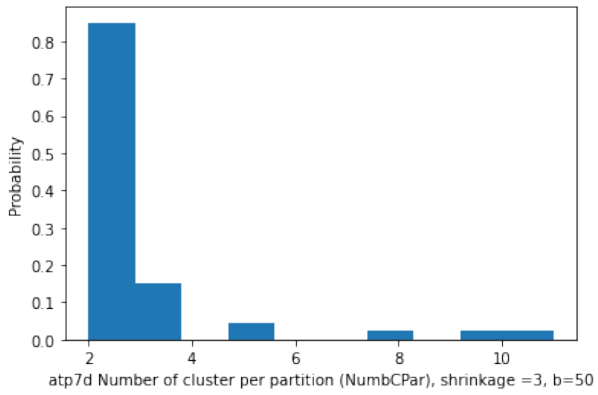


(a) $b = 50$

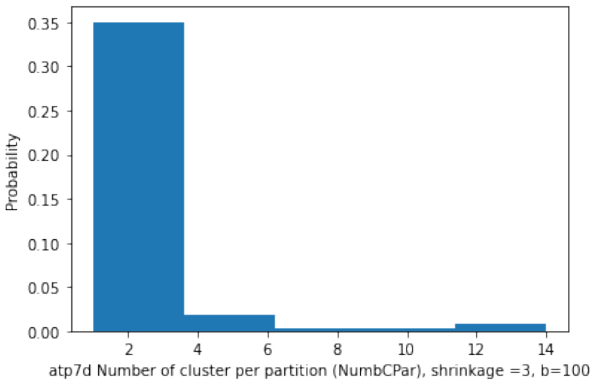


(b) $b = 100$

Figure 6: Atp7d, Number of data per cluster (NumbDC), shrinkage $= 3$

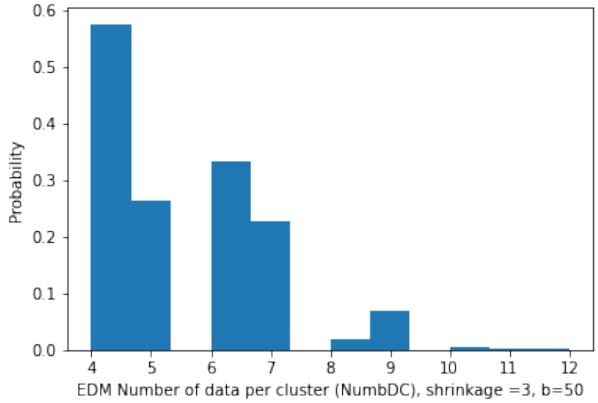


(a) $b = 50$

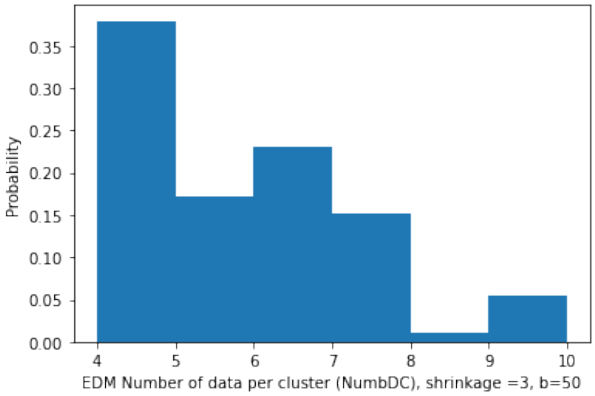


(b) $b = 100$

Figure 7: Atp1d, Number of cluster per partition (NumbCPar), shrinkage $= 3$

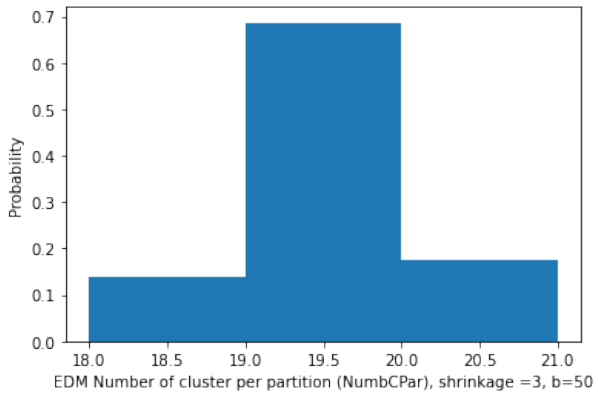


(a) $b = 50$

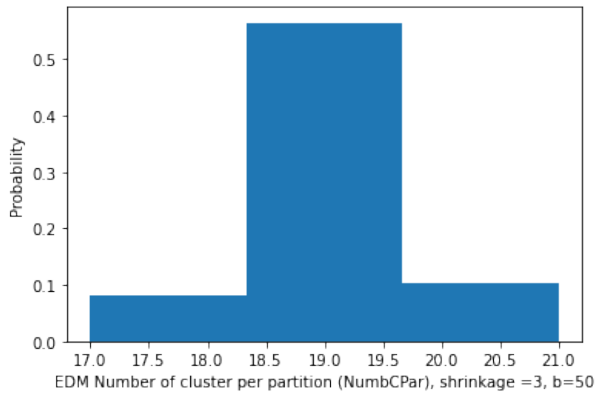


(b) $b = 100$

Figure 8: EDM, Number of data per cluster (NumbDC), shrinkage $= 3$

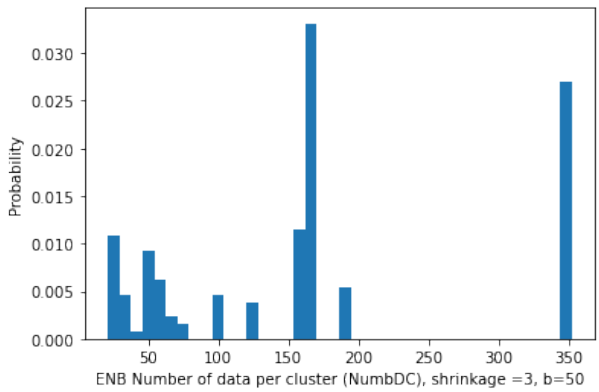


(a) $b = 50$

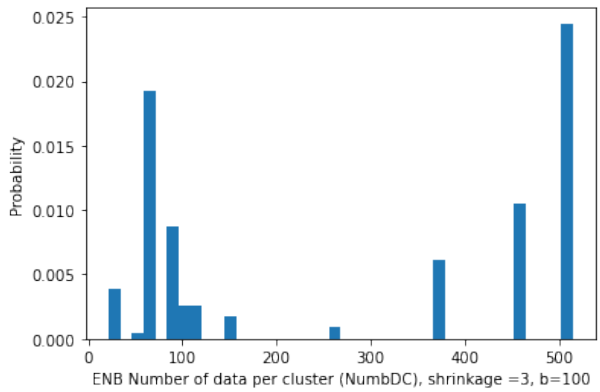


(b) $b = 100$

Figure 9: EDM, Number of cluster per partition (NumbCPar), shrinkage $= 3$

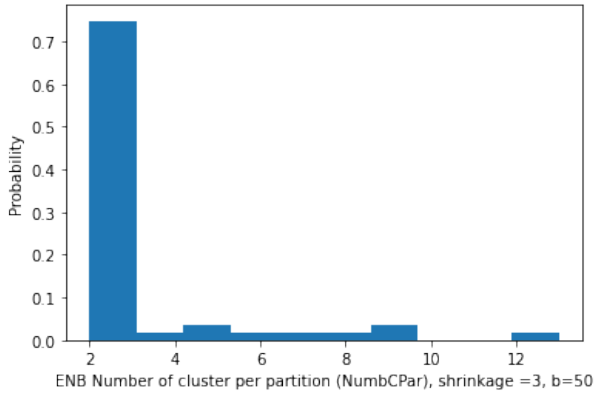


(a) $b = 50$

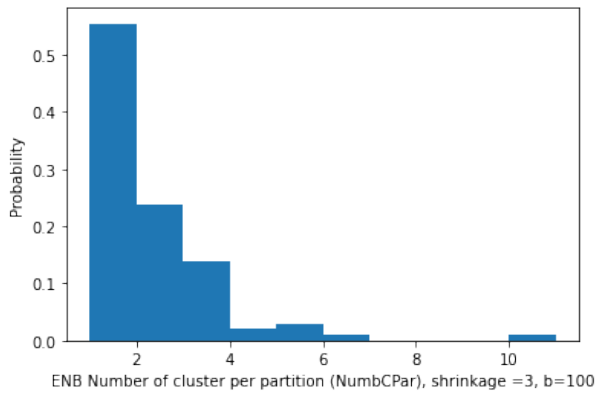


(b) $b = 100$

Figure 10: ENB, Number of data per cluster (NumbDC), shrinkage $= 3$

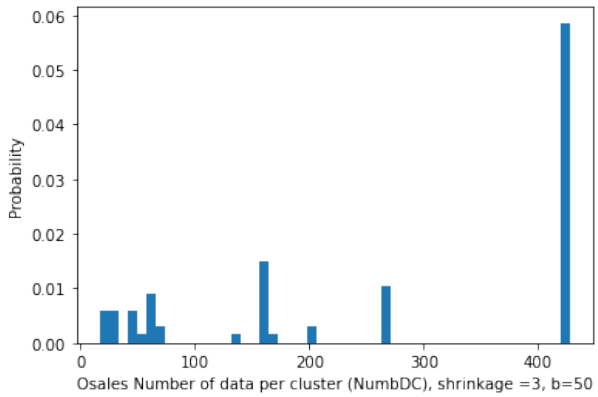


(a) $b = 50$

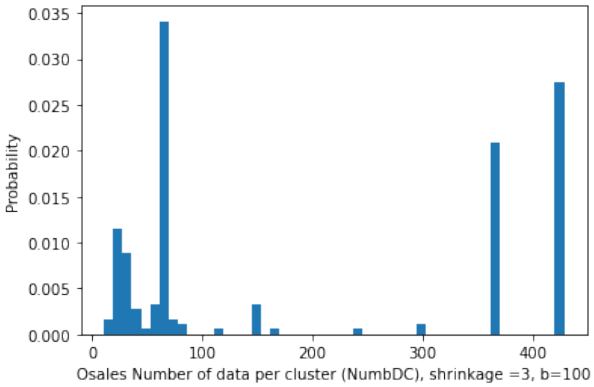


(b) $b = 100$

Figure 11: ENB, Number of cluster per partition (NumbCPar), shrinkage $= 3$

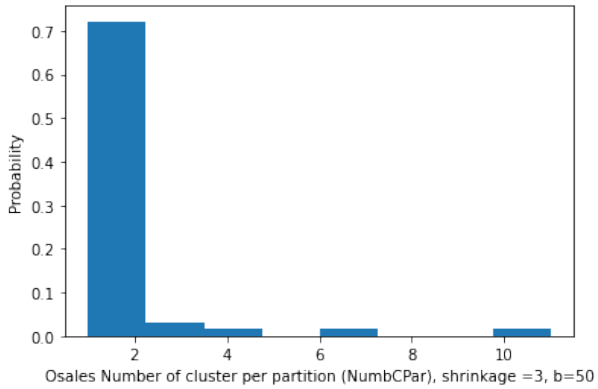


(a) $b = 50$

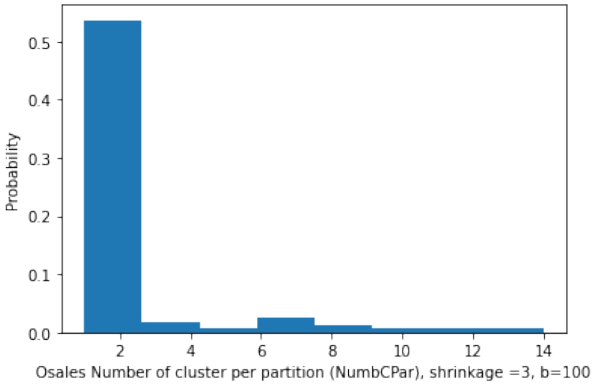


(b) $b = 100$

Figure 12: Online sales, Number of data per cluster (NumbDC), shrinkage $= 3$

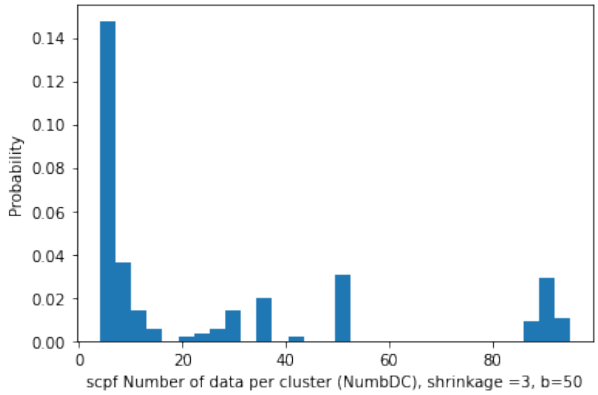


(a) $b = 50$

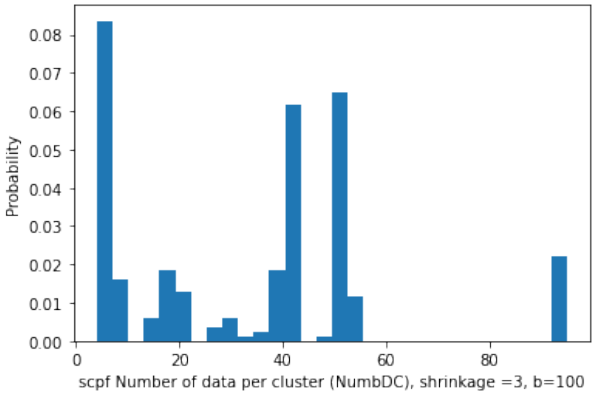


(b) $b = 100$

Figure 13: Online sales, Number of cluster per partition (NumbCPar), shrinkage $= 3$

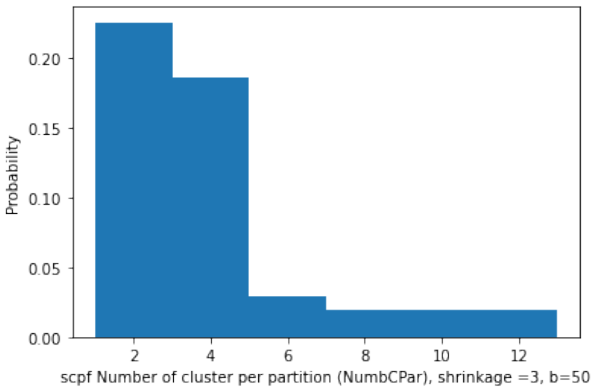


(a) $b = 50$

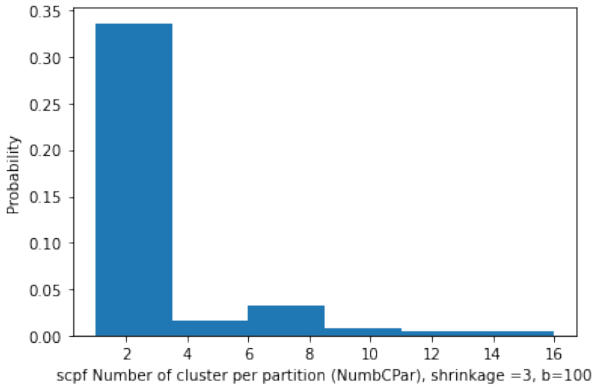


(b) $b = 100$

Figure 14: scpf, Number of data per cluster (NumbDC), shrinkage $= 3$

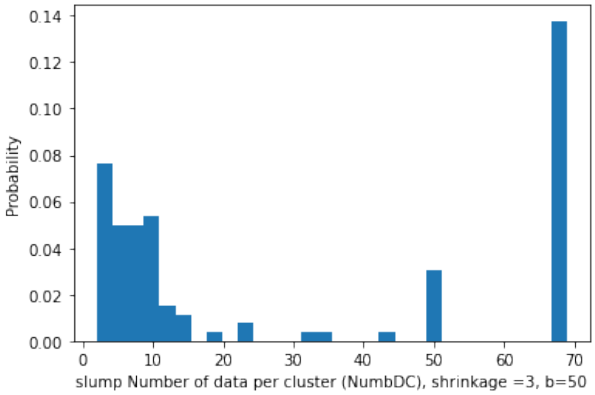


(a) $b = 50$

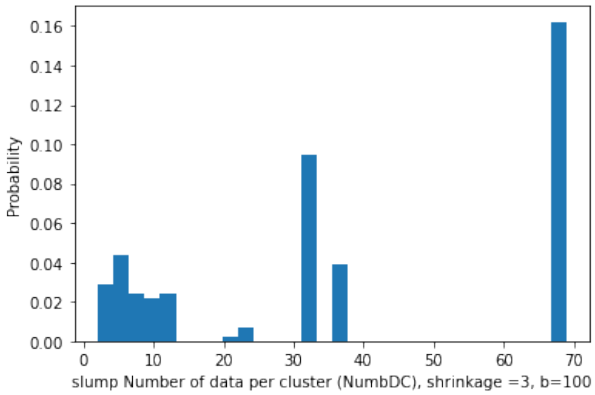


(b) $b = 100$

Figure 15: scpf, Number of cluster per partition (NumbCPar), shrinkage $= 3$

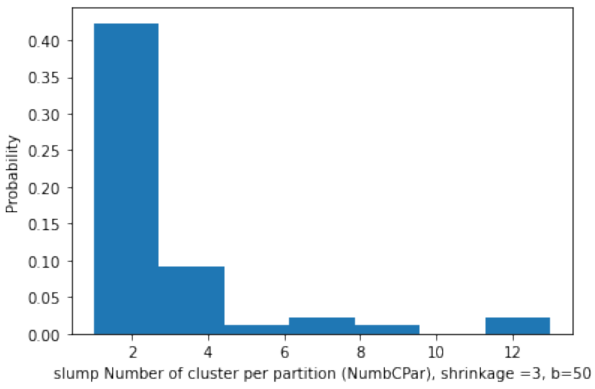


(a) $b = 50$

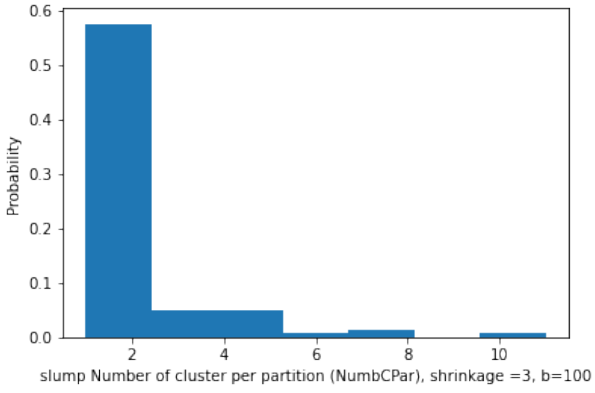


(b) $b = 100$

Figure 16: slump, Number of data per cluster (NumbDC), shrinkage $= 3$

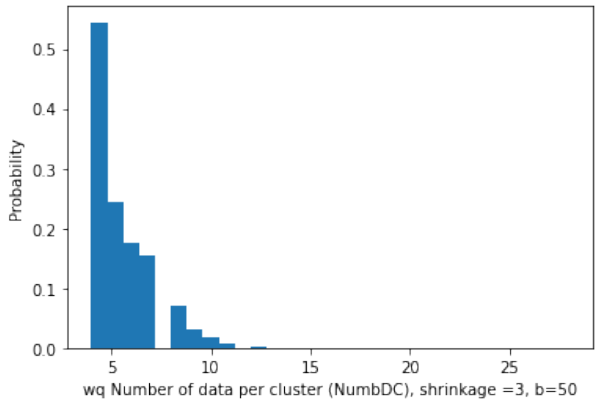


(a) $b = 50$

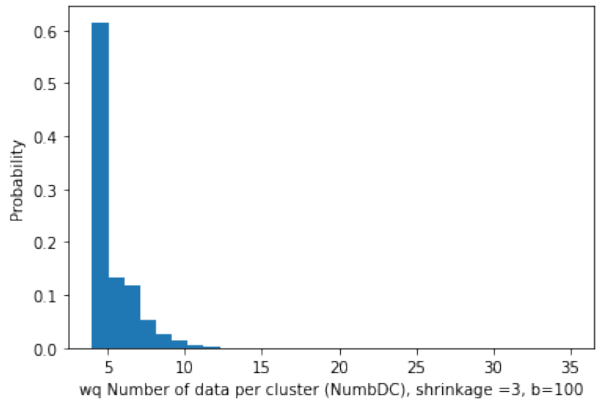


(b) $b = 100$

Figure 17: slump, Number of cluster per partition (NumbCPar), shrinkage $= 3$

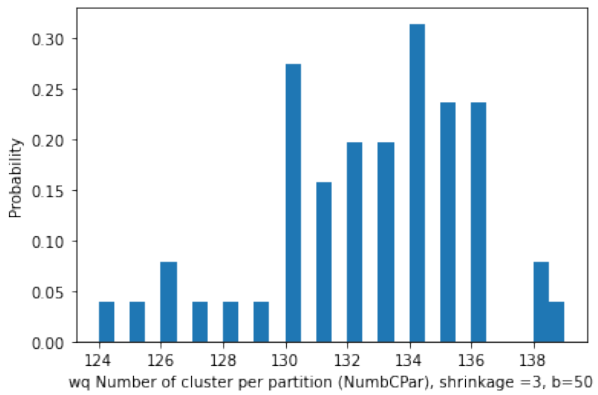


(a) $b = 50$

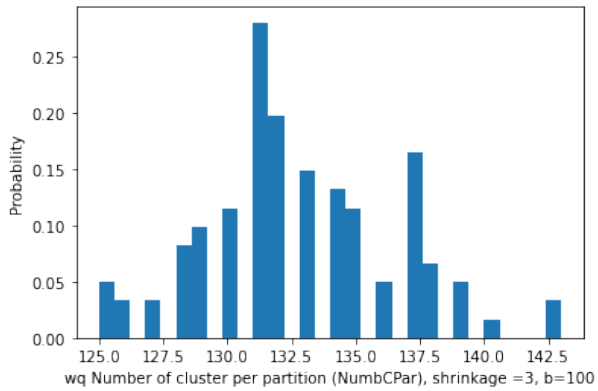


(b) $b = 100$

Figure 18: Water Quality, Number of data per cluster (NumbDC), shrinkage $= 3$



(a) $b = 50$



(b) $b = 100$

Figure 19: Water Quality, Number of cluster per partition (NumbCPar), shrinkage $= 3$

10. Extended Research on the conditional bonds distribution given the components size constraint

10.1 Frequency of frequencies distribution

The objective of the Potts clustering algorithm is to find $R(1 \leq R \leq n)$ clusters, $\mathcal{C} = \{\mathcal{C}_1, \mathcal{C}_2, ..., \mathcal{C}_R\}$. Each clustering specifies a sequence of cluster sizes, namely, letting $c_h = |C_h|$ be the size of the h-th cluster, then the sequence of cluster sizes is $[c_1, c_2, ..., c_R]$. Let denote $u_i \in \{1, ..., R\}$, the cluster observation x_i is assigned to, $c_h = \sum_{i=1}^{n} \delta(u_i = h)$ the number of observations in cluster h, and $m_v = \sum_{h=1}^{R} \delta(c_h = v)$ the number of clusters of size v, where here $\delta(x) = 1$ if condition x is satisfied, and $\delta(x) = 0$ otherwise. Thus by definition (Zhou et al., 2017), we have:

$$R = \sum_{i=1}^{n} m_i \text{ and } n = \sum_{i=1}^{n} i m_i$$

For example, if $n = 14$, $(u_1, ..., u_{14}) = (1,2,3,4,5,5,6,6,6,6,7,7,7,7)$ as $(c_1, ..., c_7) = (1,1,1,1,2,4,4)$, we have $\{m_1, m_2, m_4\} = \{4,1,2\}$ and $m_i = 0$ for $i \notin \{1,2,4\}$.

As a reminder, R is a random variable. We refer the count vector $\mathcal{M} = (\{m_v\})_v$ as the frequency of frequencies (FoF) vector, the distribution of which is commonly referred to as the FoF distribution (Good, 1953).

10.2 Objective

Our objective is to find the FoF distribution of clusters count vector given the bonds probabilities. Knowing the graph G and its edges set $E(G)$, with $\delta_{ij} = 1$ if i and j are connected, this is express as :

$$FoF(\{m_v\} | \{\delta_{ij}\}; E(G))$$

And then derive the conditional distribution of bonds given the constraint cluster size condition $\mathcal{S}_c = \{m_v = 0, \text{ for } v \leq S_c\}$, where S_c is the minimum cluster size we want.

This is express as :

$$p(b_{ij} | \{z_{si}\}, \mathcal{S}_c)$$

Remark
As highlighted above, q influences the clustering. One may be interested in :

1. *The FoF distribution $\mathcal{W} = (\{\omega_v\})_v$ for the count vector of all subsets (of data) of unique label after the first step of label assignment when q is given. This will provide a way to condition the label assignment process with constraint subset size condition $\mathcal{S}_d = \{\omega_v = 0, \text{ for } v \leq S_d\}$, where S_d is the minimum subset size we want. Let $\{\hat{z}_{si}\}$ be the conditioned label assignment.*

2. *And then search for the conditional distribution of bonds given the constraint subset size condition $\mathcal{S}_d$ and the constraint cluster size condition $\mathcal{S}_c$. This is expressed as :*

$$p(b_{ij}|\mathcal{S}_d, \mathcal{S}_c) = p(b_ij|\{\hat{z}_{si}\}, \mathcal{S}_c)$$

3. *Finally, one can insert a prior $p(\sigma, T)$ as done by Murua & Wicker (2014a) and access the MAP for the model [but not required].*

10.3 Methodology and combinatorial approach to the count vector

We present a mathematical framework to compute the distribution of the count vector $\mathcal{W} = (\{\omega_v\})_v$.

10.3.1 The FoF distribution for all subsets (of data) of unique label (at initial labelling)

Initial labelling may have a great influence on the clustering process. To find the FoF distribution for all subsets (of data) of unique label (at initial labelling) when q is given, the Potts Model can be related to many-body quantum systems problem, and describe in terms of a probability distribution over the quantum states (Tong (2012) and Tong (2006)).

Knowing the probability of a given labelling configuration $\{z_{si}\}$ as $p(\{z_{si}\}|\sigma, X, \beta, q)$, one can infer $p(\{\omega_v\}|\sigma, X, \beta, q)$. Each state configuration induces a sequence of subsets (of data) of unique label. Let $[L_1, L_2, ..., L_e]$, and $[l_1, l_2, ..., l_e]$ the sequence of subsets of data (of unique label) and their #-cardinal respectively. $z_i \in \{1, ..., e\}$ represents the subset of observation i, such that :

$$p(\{z_{si}\}|\sigma, X, \beta, q) = p(\{z_i\}|\sigma, X, \beta, q)$$

Let denote V the index set of $\{\omega_v\}$ we have necessarily $\#V \leq n$.

$$\begin{aligned} p(\{\omega_v\}|\sigma, X, \beta, q) &= p(\omega_1, \omega_2, ..., \omega_v, ..., \omega_n, \sum_v \omega_v = e|\sigma, X, \beta, q) \\ &= \sum_{Hot_w(i_i, i_2, ..., i_e)=1} p(l_1 = i_1, l_2 = i_2, ..., l_e = i_e|\sigma, X, \beta, q) \\ &= \sum_{\substack{Hot_w(i_i, i_2, ..., i_e) = 1 \\ Hat_w(z_1, z_2, ..., z_n) = 1}} p(z_1, z_2, ..., z_n|\sigma, X, \beta, q)) \end{aligned} \tag{4}$$

with

$$Hot_v^w = \left\{(i_1, i_2, ..., i_k, ..., i_e) \in \mathbf{N}^*, 1 \leq i_k \leq n, \text{ and } \sum \delta(i_k = v) = \omega_v \text{ for all } \omega_v \in (\omega_1, \omega_2, ..., \omega_v)\right\}$$

,

$$Hat_v^w = \left\{(z_1, z_2, ..., z_k, ..., z_n) \in \mathbf{N}^*, 1 \leq z_k \leq e \text{ and } \sum \delta(z_k = j) = i_j, \text{ for all } i_j \in (i_i, i_2, ..., i_e)\right\}$$

and

$$Hot_w(i_i, i_2, ..., i_e) = \begin{cases} 1 \text{ if } (i_i, i_2, ..., i_e) \in Hot_v^w \\ 0 \text{ otherwise} \end{cases}$$

$$Hat_w(z_1, z_2, ..., z_k, ..., z_n) = \begin{cases} 1 \text{ if } (z_1, z_2, ..., z_k, ..., z_n) \in Hat_v^w \\ 0 \text{ otherwise} \end{cases}$$

Based on previous computation, it is easy to get the conditioned label assignment distribution including the subset size constraint $\mathcal{S}_d$:

$$\begin{aligned} p(\{z_{si}\}|\sigma, X, \beta, q, S_d) &= p(\{z_i\}|\sigma, X, \beta, q, S_d) \\ &= \frac{p(\{z_i\}, S_d|\sigma, X, \beta, q)}{p(S_d|\sigma, X, \beta, q)} \\ &= \frac{p(\{z_i\}, S_d|\sigma, X, \beta, q)}{p(\{\omega_v = 0, \text{ for } v \leq S_d\}|\sigma, X, \beta, q)} \end{aligned} \tag{5}$$

The advantage of this conditioned distribution is that even if q is larger, subsets of significant size (size larger than in S_d constraint) can be generated at first iteration. As we know how to compute $p(\{z_i\}|\sigma, X, \beta, q)$ [by coming back to $p(\{z_{si}\}|\sigma, X, \beta, q, S_d)$] with the Hamiltonian quantity, it is easy to deduce also $p(\{z_i\}, S_d|\sigma, X, \beta, q)$. Because $\{z_i\}$ induced by $\{z_{si}\}$ give a subsets sequence configuration, it is simple to understand the combination : $\Big\{\{z_i\}, S_d\Big\}$, which is equal to $\Big\{\{z_{si}\}, \{\omega_v = 0, \text{ for } v \leq S_d\}\Big\}$ and evaluate the expression above. Of course, the condition $\Big\{\{z_{si}\}, \{\omega_v = 0, \text{ for } v \leq S_d\}\Big\}$ can be fitted by combinatorial *search* on computer if the dataset is not too large.

10.3.2 FoF distribution for the Potts clusters

The following calculations are based on section 10.1 notations. Given the symmetric matrix M_b of bonds probabilities $\{p_{ij}\}$, one can infer a combinatorial approach to get the FoF distribution of clusters count vector. For example, if $n = 3$ and M_b defined as follows :

$$M_b = \begin{pmatrix} 0 & p_{12} & p_{13} \\ p_{12} & 0 & p_{23} \\ p_{13} & p_{23} & 0 \end{pmatrix}$$

we can see that

$$p(u_1 = 1, u_2 = 1, u_3 = 2) = p_{12}(1 - p_{23})(1 - p_{13})$$

So clearly, for R clusters, the configuration $(u_1, ..., u_n)$ with $u_i \in \{1, 2, ..., R\}$ defines a set $\mathcal{P}_f$ of disjoints frozen paths configurations for which (respective) probabilities can be easily computed.

$$\begin{aligned} p(\{m_v\}|\sigma, X, \beta, q) &= p(m_1, m_2, ..., m_v, ..., m_n, \sum_v m_v = R|\sigma, X, \beta, q) \\ &= \sum_{Hot_m(i_i, i_2, ..., i_R) = 1} p(n_1 = i_1, n_2 = i_2, ..., n_R = i_R|\sigma, X, \beta, q) \\ &= \sum_{\substack{Hot_m(i_i, i_2, ..., i_R) = 1 \\ Hat_m(u_1, u_2, ..., u_n) = 1}} p(u_1, u_2, ..., u_n|\sigma, X, \beta, q)) \end{aligned} \tag{6}$$

with

$$Hot_v^m = \left\{ (i_1, i_2, ..., i_k, ..., i_R) \in \mathbf{N}^*, 1 \leq i_k \leq n, \text{ and } \sum \delta(i_k = v) = m_v \text{ for all } m_v \in (m_1, m_2, ..., m_v) \right\}$$

,

$$Hat_v^m = \left\{ (u_1, u_2, ..., u_k, ..., u_n) \in \mathbf{N}^*, 1 \leq u_k \leq R \text{ and } \sum \delta(u_k = j) = i_j, \text{ for all } i_j \in (i_i, i_2, ..., i_R) \right\}$$

and

$$Hot_w(i_i, i_2, ..., i_e) = \begin{cases} 1 \text{ if } (i_i, i_2, ..., i_e) \in Hot_v^w \\ 0 \text{ otherwise} \end{cases}$$

$$Hat_w(z_1, z_2, ..., z_k, ..., z_n) = \begin{cases} 1 \text{ if } (z_1, z_2, ..., z_k, ..., z_n) \in Hat_v^w \\ 0 \text{ otherwise} \end{cases}$$

Here, $p(u_1, u_2, ..., u_n | \sigma, X, \beta, q))$ is evaluated among the set $\mathcal{P}_f$ of all disjoints frozen paths configurations induced by $(u_1, ..., u_n)$ with $u_i \in \{1, 2, ..., R\}$.

We can then simply update the bonds distribution :

$$\begin{aligned} p(\{b_{ij}\} | \sigma, X, \beta, q, S_c) &= \frac{p(\{b_{ij}\}, S_c | \sigma, X, \beta, q)}{p(S_c | \sigma, X, \beta, q)} \\ &= \frac{p(\{b_{ij}\}, S_c | \sigma, X, \beta, q)}{p(\{m_v = 0, \text{ for } v \leq S_c\} | \sigma, X, \beta, q)} \end{aligned} \tag{7}$$

Remark

1. *The probability $p(\{b_{ij}\} | \sigma, X, \beta, q, S_c)$ is implicitly given by the matrix of bonds probabilities M_b, and its exact expression can also be obtained analytically.*
2. *The matrix of bonds probabilities M_b is mainly influenced by the label assignment, because of δ_{ij} in expression of p_{ij} ($p_{ij} = p(b_{ij} | \delta_{ij}) = \delta_{ij}(1 - \exp\{-\beta k_{ij}(\sigma)\})$). $k_{ij}(\sigma)$ is always the same for any i and j.*

From previous remark, we will describe in next section how to compute $p(\{b_{ij}\}, S_c | \sigma, X, \beta, q)$, to find a fully explicit expression of $p(\{b_{ij}\} | \sigma, X, \beta, q, S_c)$, which is here the conditional bonds distribution.

10.4 The conditional bonds distribution (given the size constraint)

10.4.1 Computation method

In previous bonds evaluation approach, we start with all p_{ij} evaluation and comparison to a given threshold (let say 0.5 for example). In this approach, we automatically select the most probable $\{b_{ij}\}$

configuration. For our context of conditional bonds, evaluation of all single p_{ij} is not *useful* to access the right frozen state that fit the cluster size constraint for all bonds.

To access $p(\{b_{ij}\}, S_c|\sigma, X, \beta, q)$, the best method here is to build a list of all $\{b_{ij}\}$ configurations (w.r.t $b_{ij} = 0$ if x_i and x_j are not neighbors) that satisfy the cluster size constraint $\mathcal{S}_c = \{m_v = 0, \text{ for } v \leq S_c\}$ (by a combinatorial search), and choose the most highly probable configuration among those selected given a label assignement.

1. There is a fast computation algorithm to find the list of all bonds $\{b_{ij}\}$ configurations (w.r.t $b_{ij} = 0$ if x_i and x_j are not neighbors) that meet the cluster size constraint $\mathcal{S}_c$. This list will be called the S_c-list, and the list of all bonds $\{b_{ij}\}$ configurations (w.r.t $b_{ij} = 0$ if x_i and x_j are not neighbors) will be called the $G-$list (where notation G is set to signify : general or global).

2. The S_c-list should be built at the beginning, and the respective probability of each of its element evaluate for each label assignment (as a reminder matrix M_b changes for each label assignment).

10.5 Fast-Algorithm to find S_c-list

1. For each $\{b_{ij}\}$ configuration, we use the Hoshen-Kopelman (HK) algorithm (Li, 2011) to connect frozen spin pairs into a path of frozen bonds. This step helps in finding all clusters given the bonds configuration.

2. All frozen paths have many *extended*-version, and *shrunk*-version. The frozen path is said to be *extended* if it is augmented with other vertices. When the path is reduced to less number of vertices (contains fewer vertices), it is said to be *shrunk*. When the frozen path is simultaneously *shrunk* and *extended* (i.e some old vertices are replaced with newer vertices), it is said to be *mixed*.

3. Let denote $G-$-List$[i]$ the ith element from the G-list. To avoid searching for all frozen paths for all $\{b_{ij}\}$ configurations, here is an algorithm to go through the list and search for all $\{b_{ij}\}$ configurations that fit the clusters size condition:

S_c-algorithm

---

1 Starting at $G-$list[1], Go through G-list and find with HK algorithm the first configuration $G-$list$[i_k]$ for which the small frozen path $SP_f[i_k]$ has a length upper than in condition S_c; add $G-$list$[i_k]$ to S_c-list.

2 In the next step, use $SP_f[i_k]$ to make a selection of good candidates:

- Add to S_c-list all configurations that contain either a *shrunk*, *extended* or *mixed* version of $SP_f[i_k]$ that fit the cluster size condition $\mathcal{S}_c$. Remove those one from G-list.
- Delete from G-list all configurations that contains either a *shrunk* or *mixed* version of $SP_f[i_k]$ that don't fit the cluster size condition $\mathcal{S}_c$.

3 Move to the next $G-$list$[i_{k+1}]$ that fit the condition, and repeat step [2] by using $SP_f[i_{k+1}]$.

4 Repeat steps [1], [2] and [3] for any other element of $G-$list that fit the clusters size condition, until done!

10.6 Finding the most probable configuration among S_c-list given a label assignment

Here we describe a computation method to select the most probable configuration (in S_c-list) given a label assignment.

1. Most importantly, **given a label assignment**, we need to delete from S_c-list all configurations for which there exist any $\delta_{ij} = 0$, and $\big(1 - \exp\{-\beta k_{ij}(\sigma)\}\big)$ suggest $b_{ij} = 1$. In fact, we generate in practice the bonds with $\big(1 - \exp\{-\beta k_{ij}(\sigma)\}\big)$, and check for the state of δ_{ij} to confirm that the label assignment is the same.

   Let call $\tilde{S}_c$-list the adjusted S_c-list which includes this modification.

2. The Bonds matrix probabilities $M_b = (p_{ij})$ with $p_{ij} = p(b_{ij}|\delta_{ij}) = \delta_{ij}\big(1 - \exp\{-\beta k_{ij}(\sigma)\}\big)$, can be seen as a upper triangular matrix re-written as **an element by element** product of two upper triangular matrix with **diagonal elements set to zero** :

   $$M_b = \tilde{\mathbf{P}} \circ \tilde{\delta}$$

   with $\tilde{\mathbf{P}} = (\tilde{p}_{ij})$, $\tilde{p}_{ij} = 1 - \exp\{-\beta k_{ij}(\sigma)\}$, and $\tilde{\delta} = (\delta_{ij})$. Only matrix $\tilde{\delta}$ change after each label assignment.

   The upper triangular bonds matrix $B = (b_{ij})$ with **diagonal elements set to zero** can also be introduced for a given configuration $\{b_{ij}\}$ in $\tilde{S}_c$-list.

   The probability of a given configuration in $\tilde{S}_c$-list is the product of all elements in upper right-corner of the following triangular matrix (excluding diagonal elements):

   $$[\tilde{P} \circ \tilde{\delta}]^B + [(1 - \tilde{P}) \circ \tilde{\delta}]^{1-B}$$

   for which each non-null element is computed as :

   $$\big(\tilde{p}_{ij} \cdot \delta_{ij}\big)^{b_{ij}} + \big(\{1 - \tilde{p}_{ij}\} \cdot \delta_{ij}\big)^{1-b_{ij}}$$

   We have then :

   $$p(\{b_{ij}\}, \{b_{ij}\} \in \tilde{S}_c - \text{list}|\sigma, X, \beta, q, S_c) \quad = \quad \frac{PROD_{i\in\{1,\dots,n-1\},j>i}\Big([\tilde{P} \circ \tilde{\delta}]^B + [(1 - \tilde{P}) \circ \tilde{\delta}]^{1-B}\Big)}{p(\{m_v = 0, \text{ for } v \leq S_c\}|\sigma, X, \beta, q)} \tag{8}$$

   where $PROD_{i\in\{1,\dots,n-1\},j>i}$ describe the product of all elements in the upper right-corner of the matrix.

3. The most probable configuration can then be selected based on this computation. This probability in 8 is computationally intractable for large datasets but remains a contribution of this research, as it can be used for further calculus, or mathematics developments.

11. Conclusion and final notes

11.1 Notes on the Potts Models and the constrained bonds distribution.

The Potts model is frequently used as random partitions models where covariates may be included [See 3]. Those models with covariates are called Product partition models with a covariate-dependent extension (PPMx) mainly introduced by Müller et al. (2011). As for example, the Shallow Gibbs Model (ALAHASSA & Murua, 2021) is based on a PPMx from Murua & Quintana (2017b) which has the form:

$$y_1, \ldots, y_n \mid \rho_n, \boldsymbol{\psi}_1^*, \ldots, \boldsymbol{\psi}_{k_n}^* \overset{\text{ind}}{\sim} p\left(\boldsymbol{y}_i \mid \boldsymbol{x}_i, \boldsymbol{\psi}_{s_i}^*\right)$$
$$\boldsymbol{\psi}_1^*, \ldots, \boldsymbol{\psi}_{k_n}^* \overset{\text{iid}}{\sim} p(\boldsymbol{\psi}) \text{ and } \rho_n \sim p\left(\rho_n \mid \boldsymbol{x}^n\right)$$

where $p\left(y_i \mid x_i, \psi_i\right)$ is the likelihood model stating the relationship between the i th response and the associated covariate vector x_i. The Potts clusters in this model are simulated using the Swendsen-Wang algorithm (Wang & Swendsen (1990), Barbu & Zhu (2003), Galanis et al. (2019)). One may be confronted to the cluster size distribution during simulation, because the *bonds* based version of the Potts Model [see 2.1] has the drawback to simulate small clusters that are not preferable in some circumstances, as for example to run or apply some neural networks models locally on the clusters themselves (using cluster adaptive training scheme as a specific example – Gales (2001)). One theoretical approach is to search for the distribution of the components size in the Potts Model to simulate a conditional distribution: a bonds distribution with constraints or restrictions on the clusters minimum size. This can be done using the frequency of frequencies distribution [see 10.1], specifically the count vector $\mathcal{M} = (\{m_v\})_v$ of the clusters also called the frequency of frequencies (FoF) vector, the distribution of which is also commonly referred to as the FoF distribution in short appellation. Once you find the FoF distribution of clusters count vector given the bonds probabilities, expressed as :

$$FoF(\{m_v\}|\{\delta_{ij}\}; E(G))$$

with the graph G and its edges set $E(G)$ ($\delta_{ij} = 1$ if observations i and j are connected), you may derive the conditional distribution of bonds given the constraint cluster size condition $\mathcal{S}_c = \{m_v = 0, \text{ for } v \leq S_c\}$, where S_c is the minimum cluster size we expect. This requires that we explore all configurations of $\mathcal{S}_c$ [see 10.4.1], for which we have proposed a fast algorithm. This may almost be intractable for our usual computers (only for large datasets), even to find the most probable configuration among S_c-list given a label assignment. We have just limited this research to simply present this conditional bonds distribution we were looking for in [see 10.6]:

$$p(\{b_{ij}\}, \{b_{ij}\} \in \tilde{S}_c - \text{list}|\sigma, X, \beta, q, S_c) = \frac{PROD_{i\in\{1,\ldots,n-1\}, j>i}\left([\tilde{P} \circ \tilde{\delta}]^B + [(1-\tilde{P}) \circ \tilde{\delta}]^{1-B}\right)}{p(\{m_v = 0, \text{ for } v \leq S_c\}|\sigma, X, \beta, q)} \tag{9}$$

where we are given a q-state, a bandwidth σ, X is set for our covariates, and β the inverse of the system temperature, and the other parameters are reformulated quantities presented in section 10.

11.2 The Potts Model with Complete Shrinkage:

Another practical approach is to modify slightly the Swendsen-Wang algorithm to insert some clusters constraints. This what we have achieved as a novel Potts Model, called the Potts Clustering with Complete Shrinkage (PCCS) [see 6.1]. In this approach, to deal with the increasing number of small clusters generated in a given partition, we have applied a modified agglomerative clustering approach

(Kurita, 1991) by merging all small clusters of size $\leq$ **h** with their closest cluster in terms of minimal distance respectively, where **h** is an integer greater or equal to 2. The algorithm uses a technique in which distances of all pairs of observations are stored. Then the nearest cluster (with size $\geq$ **h**) is given by the cluster with the closest node in terms of minimal distance to the cluster to be merged. This approach is truly effective; it helps to control the clusters size, and we have found evidence of Chi-Square and Gamma density curves [figure 2, figure 18] of the constrained cluster size distribution of PCCS, when applied to some datasets taken from the multiple-output benchmark datasets available in the Mulan project website (Tsoumakas et al., 2020) [see Table 1].

11.3 On further releases of the software package *pottscompleteshrinkage*

As we are still working on the package built-in methods to include more utilities, we expect to add in further releases a possibility to choose other kernels [5], and some Python classes to estimate the appropriate/optimal value for each parameter (mainly for the temperature T, the number of states q, the bandwidth σ). Advanced releases and code snippets of the package will be published in Journal of Statistical Software (JSS).

11.3.1 On the possibility to include other algorithms at the Shrinkage step:

It is also possible to develop many other clustering shrinkage-based model, which can involve various types of linkage [6], as well as other ***types of models*** to build different constraints or algorithms/methods:

1. Minimum and Maximum Size, or Same Size constraints : those algorithms do exist, and can be copied and remodelled for implementations in the Potts model as well. For example, one can remodel the following algorithms: Same Size Contrained KMeans Heuristics, Minimum and Maximum Size Constrained KMeans (Basu et al. (2008), Ng (2000), Bhattacharya et al. (2018), Bradley et al. (2000), Tzortzis & Likas (2014), Malinen & Fränti (2014)).

2. Annealling Algorithms for the Potts Model: those methods focus on the energy of the system to return the correspondent clusters (Tsuchiya et al. (2001), Wersing & Ritter (1999), Okada et al. (2020), Monsivais-Alonso (2006), Rose et al. (1991), Baranwal & Salapaka (2017)).

3. Pitman-Yor Process can also be used to run partionning of data, as it can detect the right number of components : Fan & Bouguila (2019), Chatzis (2013), Knowles & Ghahramani (2014), Qiang et al. (2018). But, this method requires rigorous application as it is a probabilistic method that may have some inconsistency in practice Miller & Harrison (2014). Not the least, when constraining the number of components with a Pitman-Yor Process, Dirichelet Process are useful to constrain each cluster size: Li et al. (2016), Vlachos et al. (2009), Klami & Jitta (2016), Jitta & Klami (2018).

12. Acknowledgments

The author would like to thank its Professor Alejandro Murua for his substantive contribution and all the administration staff from the Head Office of the Department of Mathematics and Statistics of University of Montreal, for their careful support while writing the manuscript. My Special recognition is also given to Professor Marlène Frigon (Director & Chair of the department) and Professor Pierre Duchesne (Head of Statistics Program). This research was supported by the Institute for Data Valorization (IVADO) of Canada.

References

Abdollah-nia, M.-F., Saeedghalati, M., & Abbassian, A. (2012). Optimal region of latching activity in an adaptive potts model for networks of neurons. *Journal of Statistical Mechanics: Theory and Experiment*, *2012*(02), P02018.

Ackerman, M., Ben-David, S., & Loker, D. (2010). Characterization of linkage-based clustering. In *COLT*, (pp. 270–281).

Ackermann, M. R., Blömer, J., Kuntze, D., & Sohler, C. (2014). Analysis of agglomerative clustering. *Algorithmica*, *69*(1), 184–215.

Agarwal, D. K. (2002). Shrinkage estimator generalizations of proximal support vector machines. In *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, (pp. 173–182).

Ahmad, M. A. K., Liao, L., & Saburov, M. (2018). Periodic p-adic gibbs measures of q-state potts model on cayley trees i: The chaos implies the vastness of the set of p-adic gibbs measures. *Journal of Statistical Physics*, *171*(6), 1000–1034.

Ahmed, H. A., Mahanta, P., Bhattacharyya, D., & Kalita, J. K. (2011). Gerc: tree based clustering for gene expression data. In *2011 IEEE 11th International Conference on Bioinformatics and Bioengineering*, (pp. 299–302). IEEE.

Ahmed, S. (1997). Asymptotic shrinkage estimation: the regression case. *Applied Statistical Science II*, (pp. 113–139).

Ahmed, S. E. (2014). *Penalty, shrinkage and pretest strategies: variable selection and estimation*. Springer.

Ahmed, S. E., & Nicol, C. J. (2012). An application of shrinkage estimation to the nonlinear regression model. *Computational Statistics & Data Analysis*, *56*(11), 3309–3321.

Aiyer, S. V., Niranjan, M., & Fallside, F. (1990). A theoretical investigation into the performance of the hopfield model. *IEEE transactions on neural networks*, *1*(2), 204–215.

ALAHASSA, N. K.-A., & Murua, a. (2021). The shallow gibbs network, double backpropagation and differential machine learning. *ScienceOpen Preprints*.

Albert, P. J., & Schwarz, U. S. (2014). Dynamics of cell shape and forces on micropatterned substrates predicted by a cellular potts model. *Biophysical journal*, *106*(11), 2340–2352.

Alexandru, A., Bergner, G., Schaich, D., & Wenger, U. (2018). Solution of the sign problem in the potts model at fixed fermion number. *Physical Review D*, *97*(11), 114503.

Anderberg, M. R. (2014). *Cluster analysis for applications: probability and mathematical statistics: a series of monographs and textbooks*, vol. 19. Academic press.

Andes, N. (1998). Robust strategies for application of cluster analyses. *Bulletin of Sociological Methodology/Bulletin de Méthodologie Sociologique*, *58*(1), 31–47.

Antoniak, C. E. (1974). Mixtures of dirichlet processes with applications to bayesian nonparametric problems. *The annals of statistics*, (pp. 1152–1174).

Aquino, É. O., Lima, F., Araújo, A. D., & Costa Filho, R. N. (2018). Potts model in one-dimension on directed small-world networks. *Journal of Statistical Physics*, *171*(6), 1112–1121.

Ashkin, J., & Teller, E. (1943). Statistics of two-dimensional lattices with four components. *Physical Review*, *64*(5-6), 178.

Asikainen, J., Aharony, A., Mandelbrot, B., Rausch, E., & Hovi, J.-P. (2003). Fractal geometry of critical potts clusters. *The European Physical Journal B-Condensed Matter and Complex Systems*, *34*(4), 479–487.

AZIZI, E., KHARRATI-SHISHAVAN, H., MOHAMMADI-IVATLOO, B., & SHOTORBANI, A. M. (2019). Wind speed clustering using linkage-ward method: A case study of khaaf, iran. *Gazi University Journal of Science*, *32*(3), 945–954.

Bai, L., Cheng, X., Liang, J., & Guo, Y. (2017). Fast graph clustering with a new description model for community detection. *Information Sciences*, *388*, 37–47.

Baranwal, M., & Salapaka, S. M. (2017). Clustering with capacity and size constraints: A deterministic approach. In *2017 Indian Control Conference (ICC)*, (pp. 251–256). IEEE.

Barbu, A., & Zhu, S.-C. (2003). Graph partition by swendsen-wang cuts. In *null*, (p. 320). IEEE.

Barcella, W., De Iorio, M., & Baio, G. (2017). A comparative review of variable selection techniques for covariate dependent dirichlet process mixture models. *Canadian Journal of Statistics*, *45*(3), 254–273.

Barry, D., & Hartigan, J. A. (1992). Product partition models for change point problems. *The Annals of Statistics*, (pp. 260–279).

Basu, S., Davidson, I., & Wagstaff, K. (2008). *Constrained clustering: Advances in algorithms, theory, and applications*. CRC Press.

Baxter, R. J. (1973). Potts model at the critical temperature. *Journal of Physics C: Solid State Physics*, *6*(23), L445.

Berganza, M. I., Petri, A., & Coletti, P. (2014). Dynamic metastability in the two-dimensional potts ferromagnet. *Physical Review E*, *89*(5), 052115.

Betancourt, B., Zanella, G., & Steorts, R. C. (2020). Random partition models for microclustering tasks. *arXiv preprint arXiv:2004.02008*.

Bhattacharya, A., Jaiswal, R., & Kumar, A. (2018). Faster algorithms for the constrained k-means problem. *Theory of computing systems*, *62*(1), 93–115.

Biehler, R. (1997). Software for learning and for doing statistics. *International Statistical Review*, *65*(2), 167–189.

Bisconti, C., Corallo, A., Fortunato, L., Gentile, A. A., Massafra, A., & Pellè, P. (2015). Reconstruction of a real world social network using the potts model and loopy belief propagation. *Frontiers in psychology*, *6*, 1698.

Blackwell, D., MacQueen, J. B., et al. (1973). Ferguson distributions via pólya urn schemes. *The annals of statistics*, *1*(2), 353–355.

Blashfield, R. K. (1980). The growth of cluster analysis: Tryon, ward, and johnson. *Multivariate behavioral research*, *15*(4), 439–458.

Blashfield, R. K., & Aldenderfer, M. S. (1988). The methods and problems of cluster analysis. In *Handbook of multivariate experimental psychology*, (pp. 447–473). Springer.

Blatt, M., Wiseman, S., & Domany, E. (1996a). Clustering data through an analogy to the potts model. In *Advances in Neural Information Processing Systems*, (pp. 416–422).

Blatt, M., Wiseman, S., & Domany, E. (1996b). Superparamagnetic clustering of data. *Physical review letters*, *76*(18), 3251.

Blatt, M., Wiseman, S., & Domany, E. (1996c). Superparamagnetic clustering of data. *Physical Review Letters*, *76*, 3251–3254.

Blatt, M., Wiseman, S., & Domany, E. (1997). Data clustering using a model granular magnet. *Neural Computation*, *9*(8), 1805–1842.

Blei, D. M., & Frazier, P. I. (2011). Distance dependent chinese restaurant processes. *Journal of Machine Learning Research*, *12*(8).

Borgs, C., Chayes, J. T., & Tetali, P. (2012). Tight bounds for mixing of the swendsen–wang algorithm at the potts transition point. *Probability Theory and Related Fields*, *152*(3-4), 509–557.

Bradley, P. S., Bennett, K. P., & Demiriz, A. (2000). Constrained k-means clustering. *Microsoft Research, Redmond*, *20*(0), 0.

Branco, N., & Girardi, D. (2012). Numerical study of potts models with aperiodic modulations: influence on first-order transitions. In *APS March Meeting Abstracts*, vol. 2012, (pp. D52–009).

Breger, A., Ehler, M., Bogunovic, H., Waldstein, S., Philip, A.-M., Schmidt-Erfurth, U., & Gerendas, B. (2017). Supervised learning and dimension reduction techniques for quantification of retinal fluid in optical coherence tomography images. *Eye*, *31*(8), 1212–1220.

Brush, S. G. (1967). History of the lenz-ising model. *Reviews of modern physics*, *39*(4), 883.

Buttrey, S. E., & Whitaker, L. R. (2015). treeclust: An r package for tree-based clustering dissimilarities. *R Journal*, *7*(2).

Camastra, F., & Verri, A. (2005). A novel kernel method for clustering. *IEEE transactions on pattern analysis and machine intelligence*, *27*(5), 801–805.

Caparica, A., Leão, S. A., & DaSilva, C. J. (2015). Static critical behavior of the q-states potts model: High-resolution entropic study. *Physica A: Statistical Mechanics and its Applications*, *438*, 447–453.

Cardy, J., & Ziff, R. M. (2003). Exact results for the universal area distribution of clusters in percolation, ising, and potts models. *Journal of statistical physics*, *110*(1-2), 1–33.

Carlucci, D. M. (1999). Comment on "glassy potts model: A disordered potts model without a ferromagnetic phase". *Physical Review B*, *60*(13), 9862.

Chair, N. (2014). Trigonometrical sums connected with the chiral potts model, verlinde dimension formula, two-dimensional resistor network, and number theory. *Annals of Physics*, *341*, 56–76.

Chang, S.-C., & Shrock, R. (2000). Ground-state entropy of the potts antiferromagnet with next-nearest-neighbor spin-spin couplings on strips of the square lattice. *Physical Review E*, *62*(4), 4650.

Chatzis, S. P. (2013). A markov random field-regulated pitman–yor process prior for spatially constrained data clustering. *Pattern recognition*, *46*(6), 1595–1603.

Chennuru Vankadara, L., & Ghoshdastidar, D. (2019). On the optimality of kernels for high-dimensional clustering. *arXiv e-prints*, (pp. arXiv–1912).

Cickovski, T., Aras, K., Swat, M., Merks, R. M., Glimm, T., Hentschel, H. G. E., Alber, M. S., Glazier, J. A., Newman, S. A., & Izaguirre, J. A. (2007). From genes to organisms via the cell: a problem-solving environment for multicellular development. *Computing in science & engineering*, *9*(4), 50–60.

Cipra, B. A. (1987). An introduction to the ising model. *The American Mathematical Monthly*, *94*(10), 937–959.

Cipra, B. A. (2000). The ising model is np-complete. *SIAM News*, *33*(6), 1–3.

Coniglio, A., & Peruggi, F. (1982). Clusters and droplets in the q-state potts model. *Journal of Physics A: Mathematical and General*, *15*(6), 1873.

Dahl, D. B. (2008). Distance-based probability distribution for set partitions with applications to bayesian nonparametrics. *JSM Proceedings. Section on Bayesian Statistical Science, American Statistical Association, Alexandria, Va*.

Dahl, D. B., Day, R., & Tsai, J. W. (2017). Random partition distribution indexed by pairwise information. *Journal of the American Statistical Association*, *112*(518), 721–732.

Dahl, D. B., et al. (2009). Modal clustering in a class of product partition models. *Bayesian Analysis*, *4*(2), 243–264.

Dasgupta, C., & Pandit, R. (1987). Testing approximate theories of first-order phase transitions on the two-dimensional potts model. *Journal of statistical physics*, *47*(3), 375–396.

Dawyndt, P., De Meyer, H., & De Baets, B. (2005). The complete linkage clustering algorithm revisited. *Soft Computing*, *9*(5), 385–392.

Day, W. H., & Edelsbrunner, H. (1984). Efficient algorithms for agglomerative hierarchical clustering methods. *Journal of classification*, *1*(1), 7–24.

de Berganza, M. I. (2009). Metastability and nucleation in the 2d-potts ferromagnet. In *American Institute of Physics Conference Series*, vol. 1091, (pp. 278–279).

De Hoon, M. J., Imoto, S., Nolan, J., & Miyano, S. (2004). Open source clustering software. *Bioinformatics*, *20*(9), 1453–1454.

Di Benedetto, G., Caron, F., & Teh, Y. W. (2017). Non-exchangeable random partition models for micro-clustering. *arXiv preprint arXiv:1711.07287*.

Ding, C., Fu, Z., Guo, W., & Wu, F. (2010). Critical frontier of the potts and percolation models on triangular-type and kagome-type lattices. ii. numerical analysis. *Physical Review E*, *81*(6), 061111.

Ding, C., Wang, Y., & Li, Y. (2012). Potts and percolation models on bowtie lattices. *Physical Review E*, *86*(2), 021125.

Ding, L., & Barbu, A. (2015). Scalable subspace clustering with application to motion segmentation. *Current Trends in Bayesian Methodology with Applications*, (p. 267).

Dixit, R., & Naskar, R. (2019). Region duplication detection in digital images based on centroid linkage clustering of key–points and graph similarity matching. *Multimedia Tools and Applications*, *78*(10), 13819–13840.

Dubail, J., Jacobsen, J. L., & Saleur, H. (2010). Critical exponents of domain walls in the two-dimensional potts model. *Journal of Physics A: Mathematical and Theoretical*, *43*(48), 482002.

Duminil-Copin, H. (2016). *Graphical representations of lattice spin models: cours Peccot, College de France : janvier-fevrier 2015*. Spartacus IDH.
URL https://books.google.ca/books?id=OPFbswEACAAJ

Duminil-Copin, H. (2017). Lectures on the ising and potts models on the hypercubic lattice. In *PIMS-CRM Summer School in Probability*, (pp. 35–161). Springer.

Duminil-Copin, H., Sidoravicius, V., & Tassion, V. (2017). Continuity of the phase transition for planar random-cluster and potts models with $1 \leq q \leq 4$. *Communications in Mathematical Physics*, *349*(1), 47–107.

Dunson, D. B., & Park, J.-H. (2008). Kernel stick-breaking processes. *Biometrika*, *95*(2), 307–323.

Durand, M., & Guesnet, E. (2016). An efficient cellular potts model algorithm that forbids cell fragmentation. *Computer Physics Communications*, *208*, 54–63.

Efron, B. (1992). Introduction to james and stein (1961) estimation with quadratic loss. In *Breakthroughs in statistics*, (pp. 437–442). Springer.

Ekeberg, M., Lövkvist, C., Lan, Y., Weigt, M., & Aurell, E. (2013). Improved contact prediction in proteins: using pseudolikelihoods to infer potts models. *Physical Review E*, *87*(1), 012707.

Emmendorfer, L. R., & de Paula Canuto, A. M. (2021). A generalized average linkage criterion for hierarchical agglomerative clustering. *Applied Soft Computing*, *100*, 106990.

Fan, J., et al. (1991). On the estimation of quadratic functionals. *The Annals of Statistics*, *19*(3), 1273–1294.

Fan, W., & Bouguila, N. (2019). Simultaneous clustering and feature selection via nonparametric pitman–yor process mixture models. *International Journal of Machine Learning and Cybernetics*, *10*(10), 2753–2766.

Fasulo, D. (1999). An analysis of recent work on clustering algorithms. Tech. rep., Technical report.

Feng, D. (????). An introduction to pottsutils.

Feng, D. (2008). *Bayesian hidden Markov normal mixture models with application to MRI tissue classification*. The University of Iowa.

Feng, D., & Tierney, L. (2014). Pottsutils: Utility functions of the potts models. *R Package Version 0.3*, *2*.

Ferguson, T. S. (1973). A bayesian analysis of some nonparametric problems. *The annals of statistics*, (pp. 209–230).

Ferraz, C. H. A., & Lima, J. L. S. (2017). Three-state potts model on non-local directed small-world lattices. *Physica A: Statistical Mechanics and Its Applications*, *484*, 488–498.

Ferreira, J. A., Loschi, R. H., & Costa, M. A. (2014). Detecting changes in time series: A product partition model with across-cluster correlation. *Signal processing*, *96*, 212–227.

Ferrenberg, A. M., & Swendsen, R. H. (1988). New monte carlo technique for studying phase transitions. *Physical review letters*, *61*(23), 2635.

Fortuin, C. M., & Kasteleyn, P. W. (1972). On the random-cluster model: I. introduction and relation to other models. *Physica*, *57*(4), 536–564.

Fraley, C., & Raftery, A. E. (1998). How many clusters? which clustering method? answers via model-based cluster analysis. *The computer journal*, *41*(8), 578–588.

Fraley, C., & Raftery, A. E. (1999). Mclust: Software for model-based cluster analysis. *Journal of classification*, *16*(2), 297–306.

Fraley, C., & Raftery, A. E. (2002). Model-based clustering, discriminant analysis, and density estimation. *Journal of the American statistical Association*, *97*(458), 611–631.

Fränti, P., & Virmajoki, O. (2006). Iterative shrinking method for clustering problems. *Pattern Recognition*, *39*(5), 761–775.

Fredrickson, G. H., & Andersen, H. C. (1984). Kinetic ising model of the glass transition. *Physical review letters*, *53*(13), 1244.

Freeman, R. T. (2006). Topological tree clustering of web search results. In *International Conference on Intelligent Data Engineering and Automated Learning*, (pp. 789–797). Springer.

Freeman, R. T. (2007). Topological tree clustering of social network search results. In *International Conference on Intelligent Data Engineering and Automated Learning*, (pp. 760–769). Springer.

Friedman, A., Vasseur, R., Potter, A., & Parameswaran, S. (2017). Localization protection and symmetry breaking in one-dimensional potts chains. In *APS March Meeting Abstracts*, vol. 2017, (pp. H13–009).

Fröhlich, J., & Spencer, T. (1981). The kosterlitz-thouless transition in two-dimensional abelian spin systems and the coulomb gas. *Communications in Mathematical Physics*, *81*(4), 527–602.

Fung, G. (2001). A comprehensive overview of basic clustering algorithms.

Fung, L. H. (2012). *Model-based Clustering with Network Covariates by Combining a Modified Product Partition Model with Hidden Markov Random Field*. Ph.D. thesis, Chinese University of Hong Kong.

Fytas, N. G., Theodorakis, P. E., & Malakis, A. (2017). Interfacial adsorption in two-dimensional pure and random-bond potts models. *Physical Review E*, *95*(3), 032126.

Galanis, A., Štefankovič, D., & Vigoda, E. (2019). Swendsen-wang algorithm on the mean-field potts model. *Random Structures & Algorithms*, *54*(1), 82–147.

Gales, M. J. (2001). Multiple-cluster adaptive training schemes. In *2001 IEEE International Conference on Acoustics, Speech, and Signal Processing. Proceedings (Cat. No. 01CH37221)*, vol. 1, (pp. 361–364). IEEE.

Gandica, Y., & Chiacchiera, S. (2016). Nature of phase transitions in axelrod-like coupled potts models in two dimensions. *Physical Review E*, *93*(3), 032132.

Ganganath, N., Cheng, C.-T., & Tse, C. K. (2014). Data clustering with cluster size constraints using a modified k-means algorithm. Institute of Electrical and Electronics Engineers.

Gao, J., & Hitchcock, D. B. (2010). James–stein shrinkage to improve k-means cluster analysis. *Computational Statistics & Data Analysis*, *54*(9), 2113–2127.

Georgii, H.-O., & Häggström, O. (1996). Phase transition in continuum potts models. *Communications in mathematical physics*, *181*(2), 507–528.

Geyer, C., & Johnson, L. (2010). potts: Markov chain monte carlo for potts models. *R package version 0.4*.

Giacomin, G., & Mahfouf, R. (????). Le modele d'ising, classique et quantique. *Sujets des exposés de premiere année*, (p. 23).

Giataganas, D., Huang, C.-Y., & Lin, F.-L. (2021). Neural network flows of low q-state potts and clock models. *arXiv preprint arXiv:2102.05219*.

Girolami, M. (2002). Mercer kernel-based clustering in feature space. *IEEE Transactions on Neural Networks*, *13*(3), 780–784.

Glasbey, C. (1987). Complete linkage as a multiple stopping rule for single linkage clustering. *Journal of Classification*, *4*(1), 103–109.

Glauber, R. J. (1963). Time-dependent statistics of the ising model. *Journal of mathematical physics*, *4*(2), 294–307.

Goel, S. (2020). Learning ising and potts models with latent variables. In *International Conference on Artificial Intelligence and Statistics*, (pp. 3557–3566). PMLR.

Good, I. J. (1953). The population frequencies of species and the estimation of population parameters. *Biometrika*, *40*(3-4), 237–264.

Gorbenko, V., Rychkov, S., & Zan, B. (2018). Walking, weak first-order transitions, and complex cfts. *Journal of High Energy Physics*, *2018*(10), 1–49.

Gould, H., & Tobochnik, J. (2010). *Statistical and Thermal Physics*. Princeton University Press. URL https://www.compadre.org/stpbook/statistical-mechanics-1/ex5_6.cfm

Gower, J. C. (1967). A comparison of some methods of cluster analysis. *Biometrics*, (pp. 623–637).

Gower, J. C., & Ross, G. J. (1969). Minimum spanning trees and single linkage cluster analysis. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, *18*(1), 54–64.

Graner, F., & Glazier, J. A. (1992). Simulation of biological cell sorting using a two-dimensional extended potts model. *Physical review letters*, *69*(13), 2013.

Graves, D., & Pedrycz, W. (2007). Performance of kernel-based fuzzy clustering. *Electronics Letters*, *43*(25), 1445–1446.

Green, P. J., & Richardson, S. (2002). Hidden markov models and disease mapping. *Journal of the American statistical association*, *97*(460), 1055–1070.

Griffin, J., Brown, P., et al. (2017). Hierarchical shrinkage priors for regression models. *Bayesian Analysis*, *12*(1), 135–159.

Grimmett, G. (1994). Potts models and random-cluster processes with many-body interactions. *Journal of Statistical Physics*, *75*(1-2), 67–121.

Grimmett, G. (2004). The random-cluster model. In *Probability on discrete structures*, (pp. 73–123). Springer.

Grosswendt, A., & Roeglin, H. (2017). Improved analysis of complete-linkage clustering. *Algorithmica*, *78*(4), 1131–1150.

Gruber, M. (2017). *Improving efficiency by shrinkage: The James–stein and ridge regression estimators*. Routledge.

Guenoche, A. (1994). Spanning trees and average linkage clustering. In *New Approaches in Classification and Data Analysis*, (pp. 119–127). Springer.

Häggkvist, R., Rosengren, A., Andrén, D., Kundrotas, P., Lundow, P. H., & Markström, K. (2004). A monte carlo sampling scheme for the ising model. *Journal of statistical physics*, *114*(1-2), 455–480.

Häggström, O., Jonasson, J., Lyons, R., et al. (2002). Coupling and bernoullicity in random-cluster and potts models. *Bernoulli*, *8*(3), 275–294.

Hales, J. D., & Tikare, V. (2014). Verification and validation strategy for implementation of hybrid potts-phase field hydride modeling capability in mbm. Tech. rep., Idaho National Laboratory (INL).

Handrey Araujo Ferraz, C., & Sousa Lima, J. L. (2016). Three-state potts model on non-local directed small-world lattices. *arXiv e-prints*, (pp. arXiv–1609).

Harris, A. B., & Lubensky, T. C. (1987). Potts-model formulation of the random resistor network. *Physical Review B*, *35*(13), 6987.

Harrison, N. C., & Vasiev, B. (2008). Modelling the regression of henson's node using a modified cellular potts model.

Hartigan, J. A. (1975). *Clustering algorithms*. John Wiley & Sons, Inc.

Hartigan, J. A. (1990). Partition models. *Communications in statistics-Theory and methods*, *19*(8), 2745–2756.

Hartmann, A. K. (2005). Calculation of partition functions by measuring component distributions. *Physical review letters*, *94*(5), 050601.

He, K., Jiang, Y., & Dong, S. (2009). A hybrid parallel framework for the cellular potts model simulations. In *2009 15th International Conference on Parallel and Distributed Systems*, (pp. 624–631). IEEE.

Hennig, C., Meila, M., Murtagh, F., & Rocci, R. (2015). *Handbook of cluster analysis*. CRC Press.

Hu, C.-K. (1987). Exact cluster size distributions and mean cluster sizes for the q-state bond-correlated percolation model. *Journal of Physics A: Mathematical and General*, *20*(18), 6617.

Hu, C. W., Li, H., & Qutub, A. A. (2018). Shrinkage clustering: a fast and size-constrained clustering algorithm for biomedical applications. *BMC bioinformatics*, *19*(1), 19.

Huang, J., Sun, H., Han, J., Deng, H., Sun, Y., & Liu, Y. (2010). Shrink: a structural clustering algorithm for detecting hierarchical communities in networks. In *Proceedings of the 19th ACM international conference on Information and knowledge management*, (pp. 219–228).

Huang, J., Sun, H., Song, Q., Deng, H., & Han, J. (2012). Revealing density-based clustering structure from the core-connected tree of a network. *IEEE transactions on knowledge and data engineering*, *25*(8), 1876–1889.

Irfan, M., Javed, M., & Raza, M. A. (2013). Comparison of shrinkage regression methods for remedy of multicollinearity problem. *Middle-East Journal of Scientific Research*, *14*(4), 570–579.

Ising, E. (1925). A contribution to the theory of ferromagnetism z. *Phys*, *31*(1), 253–258.

Ising, T., Folk, R., Kenna, R., Berche, B., & Holovatch, Y. (2017). The fate of ernst ising and the fate of his model. *arXiv preprint arXiv:1706.01764*.

Izaguirre, J. A., Chaturvedi, R., Huang, C., Cickovski, T., Coffland, J., Thomas, G., Forgacs, G., Alber, M., Hentschel, G., Newman, S. A., et al. (2004). Compucell, a multi-model framework for simulation of morphogenesis. *Bioinformatics*, *20*(7), 1129–1137.

Jacobsen, J. L., Salas, J., & Scullard, C. R. (2017). Phase diagram of the triangular-lattice potts antiferromagnet. *Journal of Physics A: Mathematical and Theoretical*, *50*(34), 345002.

Jacobsen, J. L., & Saleur, H. (2008). Exact valence bond entanglement entropy and probability distribution in the x x x spin chain and the potts model. *Physical review letters*, *100*(8), 087205.

Jacobsen, J. L., & Scullard, C. R. (2013). Transfer matrix computation of critical polynomials for two-dimensional potts models. *Journal of Physics A: Mathematical and Theoretical*, *46*(7), 075001.

Jacquez, G. M. (2008). Spatial cluster analysis. *The handbook of geographic information science*, *395*(416).

Jain, A. K., Topchy, A., Law, M. H., & Buhmann, J. M. (2004). Landscape of clustering algorithms. In *Proceedings of the 17th International Conference on Pattern Recognition, 2004. ICPR 2004.*, vol. 1, (pp. 260–263). IEEE.

Janke, W., & Schakel, A. M. (2004). Geometrical vs. fortuin–kasteleyn clusters in the two-dimensional q-state potts model. *Nuclear Physics B*, *700*(1-3), 385–406.

Jarman, A. M. (2020). Hierarchical cluster analysis: Comparison of single linkage, complete linkage, average linkage and centroid linkage method.

Jiang, T., & Owen, A. B. (2003). Quasi-regression with shrinkage. *Mathematics and Computers in Simulation*, *62*(3-6), 231–241.

Jiang, Y., Kovalev, A. A., & Pryadko, L. P. (2016). Potts glass reflection of the decoding threshold for qudit quantum error correcting codes. In *APS March Meeting Abstracts*, vol. 2016, (pp. P44–009).

Jin, Y., & Xiao, F. (2013). Eliminating error accumulation in hierarchical clustering algorithms. In *2013 Fourth International Conference on Emerging Intelligent Data and Web Technologies*, (pp. 639–642). IEEE.

Jitta, A., & Klami, A. (2018). On controlling the size of clusters in probabilistic clustering. In *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 32.

Jo, S., Lee, J., Page, G., Quintana, F., Trippa, L., & Müller, P. (2015). Spatial species sampling and product partition models. In *Nonparametric Bayesian Inference in Biostatistics*, (pp. 359–375). Springer.

Johansson, J., & Pistol, M.-E. (2011). Microcanonical entropy of the infinite-state potts model. *Physics Research International*, *2011*.

Jordan, C., Livingstone, V., & Barry, D. (2007). Statistical modelling using product partition models. *Statistical Modelling*, *7*(3), 275–295.

Jothi, R., Mohanty, S. K., & Ojha, A. (2015). Fast minimum spanning tree based clustering algorithms on local neighborhood graph. In *International workshop on graph-based representations in pattern recognition*, (pp. 292–301). Springer.

Joya, G., Atencia, M., & Sandoval, F. (2002). Hopfield neural networks for optimization: study of the different dynamics. *Neurocomputing*, *43*(1-4), 219–237.

Kadowaki, T., & Nishimori, H. (1998). Quantum annealing in the transverse ising model. *Physical Review E*, *58*(5), 5355.

Katori, M. (1988). A second-order phase transition as a limit of the first-order phase transitions–coherent anomalies and critical phenomena in the potts models–. *Journal of the Physical Society of Japan*, *57*(12), 4114–4125.

Kaufman, M., & Diep, H. (2010). Hybrid defect phase transition: Renormalization group and monte carlo analysis. In *APS March Meeting Abstracts*, vol. 2010, (pp. Y13–013).

Kellom, M., & Raymond, J. (2017). Using cluster edge counting to aggregate iterations of centroid-linkage clustering results and avoid large distance matrices. *Journal of biological methods*, *4*(1).

Kim, S.-Y., & Creswick, R. J. (2001). Density of states, potts zeros, and fisher zeros of the q-state potts model for continuous q. *Physical Review E*, *63*(6), 066107.

Kimes, P. K., Liu, Y., Neil Hayes, D., & Marron, J. S. (2017). Statistical significance for hierarchical clustering. *Biometrics*, *73*(3), 811–821.

Kinzel, W., & Domany, E. (1981). Critical properties of random potts models. *Physical Review B*, *23*(7), 3421.

Klami, A., & Jitta, A. (2016). Probabilistic size-constrained microclustering. In *UAI*. Citeseer.

Knowles, D. A., & Ghahramani, Z. (2014). Pitman yor diffusion trees for bayesian hierarchical clustering. *IEEE transactions on pattern analysis and machine intelligence*, *37*(2), 271–289.

Kobe, S. (1997). Ernst ising—physicist and teacher. *Journal of statistical physics*, *88*(3), 991–995.

Kobe, S. (2000). Ernst ising 1900-1998. *Brazilian Journal of Physics*, *30*(4), 649–654.

Köhn, H.-F., & Hubert, L. J. (2014). Hierarchical cluster analysis. *Wiley StatsRef: Statistics Reference Online*, (pp. 1–13).

Kolesik, M., & Suzuki, M. (1995). Critical exponents of the 3d antiferromagnetic three-state potts model using the coherent-anomaly method. *Physica A: Statistical Mechanics and its Applications*, *216*(4), 469–477.

Komura, Y., & Okabe, Y. (2014). Cuda programs for the gpu computing of the swendsen–wang multi-cluster spin flip algorithm: 2d and 3d ising, potts, and xy models. *Computer Physics Communications*, *185*(3), 1038–1043.

Korobilis, D. (2013). Hierarchical shrinkage priors for dynamic regressions with many predictors. *International Journal of Forecasting*, *29*(1), 43–59.

Kropff, E., & Treves, A. (2005). The storage capacity of potts models for semantic memory retrieval. *Journal of Statistical Mechanics: Theory and Experiment*, *2005*(08), P08010.

Krznaric, D., & Levcopoulos, C. (1997). Optimal algorithms for complete linkage clustering in d dimensions. In *International Symposium on Mathematical Foundations of Computer Science*, (pp. 368–377). Springer.

Krznaric, D., & Levcopoulos, C. (1998). Fast algorithms for complete linkage clustering. *Discrete & Computational Geometry*, *19*(1), 131–145.

Kullmann, L., Kertesz, J., & Mantegna, R. (2000). Identification of clusters of companies in stock indices via potts super-paramagnetic transitions. *Physica A: Statistical Mechanics and its Applications*, *287*(3-4), 412–419.

Kumar, M., Kumar, R., Weigel, M., Banerjee, V., Janke, W., & Puri, S. (2018). Approximate ground states of the random-field potts model from graph cuts. *Physical Review E*, *97*(5), 053307.

Kumar, M., Kumar, R., Weigel, M., Banerjee, V., & Puri, S. (????). Approximate ground states of the random-field potts model from graph cuts and parallel tempering.

Kurita, T. (1991). An efficient agglomerative clustering algorithm using a heap. *Pattern Recognition*, *24*(3), 205–209.

Lajkó, P., & Iglói, F. (2017). Entanglement entropy of the q 4 quantum potts chain. *Physical Review E*, *95*(1), 012105.

Langone, R., Mall, R., Alzate, C., & Suykens, J. A. (2016). Kernel spectral clustering and applications. In *Unsupervised Learning Algorithms*, (pp. 135–161). Springer.

Lencsés, M., & Takács, G. (2014). Excited state tba and renormalized tcsa in the scaling potts model. *Journal of High Energy Physics*, *2014*(9), 52.

Li, C., Rana, S., Phung, D., & Venkatesh, S. (2016). Dirichlet process mixture models with pairwise constraints for data clustering. *Annals of data science*, *3*(2), 205–223.

Li, C.-D., Tan, D.-R., & Jiang, F.-J. (2018). Applications of neural networks to the studies of phase transitions of two-dimensional potts models. *Annals of Physics*, *391*, 312–331.

Li, J. (2011). *Potts model clustering for discovering patterns of epigenetic marks*. Ph.D. thesis, Rutgers University-Graduate School-New Brunswick.

Li, J. F., & Lowengrub, J. (2014). The effects of cell compressibility, motility and contact inhibition on the growth of tumor cell clusters using the cellular potts model. *Journal of theoretical biology*, *343*, 79–91.

Li, Q., Yi, F., Wang, T., Xiao, G., & Liang, F. (2017). Lung cancer pathological image analysis using a hidden potts model. *Cancer informatics*, *16*, 1176935117711910.

Lian, H. (2013). Shrinkage estimation and selection for multiple functional regression. *Statistica Sinica*, (pp. 51–74).

Lima, F. (2018). Cluster size distribution to characterize the first-order phase transitions of equilibrium systems. *Physics & Astronomy International Journal*, *2*(4), 375–376.

Lin, S., Azarnoush, B., & Runger, G. C. (2018). Crafter: a tree-ensemble clustering algorithm for static datasets with mixed attributes and high dimensionality. *IEEE Transactions on Knowledge and Data Engineering*, *30*(9), 1686–1696.

Liu, B., Xia, Y., & Yu, P. S. (2005). Clustering via decision tree construction. In *Foundations and advances in data mining*, (pp. 97–124). Springer.

Lopes, D. M., & Mombach, J. C. (2017). Two-dimensional wetting transition modeling with the potts model. *Brazilian Journal of Physics*, *47*(6), 672–677.

Lorenz, E., & Janke, W. (2007). Numerical tests of local scale invariance in ageing q-state potts models. *EPL (Europhysics Letters)*, *77*(1), 10003.

Loschi, R., & Cruz, F. (2002). An analysis of the influence of some prior specifications in the identification of change points via product partition model. *Computational Statistics & Data Analysis*, *39*(4), 477–501.

Loschi, R. H., & Cruz, F. R. (2005). Extension to the product partition model: computing the probability of a change. *Computational Statistics & Data Analysis*, *48*(2), 255–268.

Lukasová, A. (1979). Hierarchical agglomerative clustering procedure. *Pattern Recognition*, *11*(5-6), 365–381.

Lv, J.-P., Deng, Y., Jacobsen, J. L., Salas, J., & Sokal, A. D. (2018a). Duality and the universality class of the three-state potts antiferromagnet on plane quadrangulations. *Physical Review E*, *97*(4), 040104.

Lv, X., Ma, Y., He, X., Huang, H., & Yang, J. (2018b). Ccimst: A clustering algorithm based on minimum spanning tree and cluster centers. *Mathematical Problems in Engineering*, *2018*.

Machta, J., Choi, Y., Lucke, A., Schweizer, T., & Chayes, L. (1996). Invaded cluster algorithm for potts models. *Physical Review E*, *54*(2), 1332.

Madhulatha, T. S. (2012). An overview on clustering methods. *arXiv preprint arXiv:1205.1117*.

Malinen, M. I., & Fränti, P. (2014). Balanced k-means for clustering. In *Joint IAPR International Workshops on Statistical Techniques in Pattern Recognition (SPR) and Structural and Syntactic Pattern Recognition (SSPR)*, (pp. 32–41). Springer.

Mann, A. K., & Kaur, N. (2013). Review paper on clustering techniques. *Global Journal of Computer Science and Technology*.

Marinari, E., Mossa, S., & Parisi, G. (1999). Glassy potts model: A disordered potts model without a ferromagnetic phase. *Physical Review B*, *59*(13), 8401.

Martinelli, F., Olivieri, E., & Scoppola, E. (1990). Rigorous analysis of low-temperature stochastic ising models: metastability and exponential approach to equilibrium. *EPL (Europhysics Letters)*, *12*(3), 223.

Massaro, J. M. (2005). Clustering, single linkage. *Encyclopedia of Biostatistics*, *2*.

McCoy, B. M., & Wu, T. T. (2014). *The two-dimensional Ising model*. Courier Corporation.

McCullagh, P. (2011). Random permutations and partition models.

Miller, D., & Rose, K. (1994). A non-greedy approach to tree-structured clustering. *Pattern recognition letters*, *15*(7), 683–690.

Miller, J. W., & Harrison, M. T. (2014). Inconsistency of pitman-yor process mixtures for the number of components. *The Journal of Machine Learning Research*, *15*(1), 3333–3370.

Miyamoto, S., Abe, R., Endo, Y., & Takeshita, J.-I. (2015). Ward method of hierarchical clustering for non-euclidean similarity measures. In *2015 7th International Conference of Soft Computing and Pattern Recognition (SoCPaR)*, (pp. 60–63). IEEE.

Mohbey, K. K., & Thakur, G. (2013). An experimental survey on single linkage clustering. *International Journal of Computer Applications*, *76*(17), 6–11.

Monsivais-Alonso, M. d. P. (2006). Simulations in statistical physics and biology: some applications. *arXiv preprint physics/0603035*.

Moores, M., & Mengersen, K. (2018). bayesimages: Bayesian methods for image segmentation using a potts model. *R package version 0.5-3. URL https://CRAN. R-project. org/package= bayesImageS*, *12*.

Moores, M., Mengersen, K., Feng, D., & Moores, M. M. (2016). Package 'bayesimages'. *Computational Statistics & Data Analysis*, *86*, 27–41.

Moores, M., Nicholls, G., Pettitt, A., Mengersen, K., et al. (2020). Scalable bayesian inference for the inverse temperature of a hidden potts model. *Bayesian Analysis*, *15*(1), 1–27.

Müller, P., & Quintana, F. (2010). Random partition models with regression on covariates. *Journal of statistical planning and inference*, *140*(10), 2801–2808.

Müller, P., Quintana, F., & Rosner, G. L. (2011). A product partition model with regression on covariates. *Journal of Computational and Graphical Statistics*, *20*(1), 260–278.

Müller, P., Rodriguez, A., et al. (2013). Random partition models. In *Nonparametric Bayesian Inference*, (pp. 87–92). IMS and ASA.

Müllner, D. (2011). Modern hierarchical, agglomerative clustering algorithms. *arXiv preprint arXiv:1109.2378*.

Murtagh, F. (1983). A survey of recent advances in hierarchical clustering algorithms. *The computer journal*, *26*(4), 354–359.

Murtagh, F., & Legendre, P. (2014). Ward's hierarchical agglomerative clustering method: which algorithms implement ward's criterion? *Journal of classification*, *31*(3), 274–295.

Murua, A., & Quintana, F. A. (2017a). Sem iparametric bayesian regression via potts model. *Journal of Computational and Graphical Statistics*, *26*(2), 265–274.
URL https://doi.org/10.1080/10618600.2016.1172015

Murua, A., & Quintana, F. A. (2017b). Semiparametric bayesian regression via potts model. *Journal of Computational and Graphical Statistics*, *26*(2), 265–274.

Murua, A., Stanberry, L., & Stuetzle, W. (2008a). On potts model clustering, kernel k-means and density estimation. *Journal of Computational and Graphical Statistics*, *17*(3), 629–658.

Murua, A., Stanberry, L., & Stuetzle, W. (2008b). On potts model clustering, kernel k-means, and density estimation. *Journal of Computational and Graphical Statistics*, *17*(3), 629–658.

Murua, A., & Wicker, N. (2014a). The conditional-potts clustering model. *Journal of Computational and Graphical Statistics*, *23*(3), 717–739.

Murua, A., & Wicker, N. (2014b). The conditional-potts clustering model. *Journal of Computational and Graphical Statistics*, *23*(3), 717–739.
URL https://doi.org/10.1080/10618600.2013.837828

Mythili, S., & Madhiya, E. (2014). An analysis on clustering algorithms in data mining. *International Journal of Computer Science and Mobile Computing*, *3*(1), 334–340.

Naim, M., Boboeva, V., Kang, C. J., & Treves, A. (2018). Reducing a cortical network to a potts model yields storage capacity estimates. *Journal of Statistical Mechanics: Theory and Experiment*, *2018*(4), 043304.

Navarro, C. A., Canfora, F., Hitschfeld, N., & Navarro, G. (2015). Parallel family trees for transfer matrices in the potts model. *Computer Physics Communications*, *187*, 55–71.

Ng, M. K. (2000). A note on constrained k-means algorithms. *Pattern Recognition*, *33*(3), 515–519.

Nielsen, F. (2016). Hierarchical clustering. In *Introduction to HPC with MPI for Data Science*, (pp. 195–211). Springer.

Nord, A. L., Gachon, E., Perez-Carrasco, R., Nirody, J. A., Barducci, A., Berry, R. M., & Pedaci, F. (2017). Catch bond drives stator mechanosensitivity in the bacterial flagellar motor. *Proceedings of the National Academy of Sciences*, *114*(49), 12952–12957.

Okada, S., Ohzeki, M., & Tanaka, K. (2020). Efficient quantum and simulated annealing of potts models using a half-hot constraint. *Journal of the Physical Society of Japan*, *89*(9), 094801.

O'Donoghue, S., & Bray, A. (2002). Fraction of uninfected walkers in the one-dimensional potts model. *Physical Review E*, *65*(5), 051114.

Page, G. L., & Quintana, F. A. (2018). Calibrating covariate informed product partition models. *Statistics and Computing*, *28*(5), 1009–1031.

Page, G. L., Quintana, F. A., & Dahl, D. B. (2019). Spatio-temporal random partition models. *arXiv preprint arXiv:1912.11542*.

Page, R. L. (1974). Algorithm 479: A minimal spanning tree clustering method. *Communications of the ACM*, *17*(6), 321–323.

Pan, X., Zhang, Y., Chen, L., Xu, M., & Wu, Y. (2018). Finite-size behaviour of generalized susceptibilities in the whole phase plane of the potts model. *Chinese Physics C*, *42*(2), 023110.

Patel, S., Sihmar, S., & Jatain, A. (2015). A study of hierarchical clustering algorithms. In *2015 2nd International Conference on Computing for Sustainable Global Development (INDIACom)*, (pp. 537–541). IEEE.

Pelckmans, K., De Brabanter, J., Suykens, J. A., & De Moor, B. (2005). Convex clustering shrinkage. In *PASCAL Workshop on Statistics and Optimization of Clustering Workshop*.

Pfeuty, P. (1970). The one-dimensional ising model with a transverse field. *ANNALS of Physics*, *57*(1), 79–90.

Pitman, J. (1996). Some developments of the blackwell-macqueen urn scheme. *Lecture Notes-Monograph Series*, (pp. 245–267).

Polson, N. G., & Scott, J. G. (2012). Local shrinkage rules, lévy processes and regularized regression. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, *74*(2), 287–311.

QIAN, J., & Su, L. (2016). Shrinkage estimation of regression models with multiple structural changes. *Econometric Theory*, *32*(6), 1376.

Qiang, J., Li, Y., Yuan, Y., & Wu, X. (2018). Short text clustering based on pitman-yor process mixture model. *Applied Intelligence*, *48*(7), 1802–1812.

Qiu, T., & Li, Y. (2021). Enhancing in-tree-based clustering via distance ensemble and kernelization. *Pattern Recognition*, *112*, 107731.

Quintana, F. A. (2010). Linear regression with a dependent skewed dirichlet process. *Chilean Journal of Statistics*, *1*, 35–49.

Quintana, F. A., & Iglesias, P. L. (2003). Bayesian clustering and product partition models. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, *65*(2), 557–574.

Quintana, F. A., Mueller, P., Jara, A., & MacEachern, S. N. (2020). The dependent dirichlet process and related models. *arXiv preprint arXiv:2007.06129*.

R. Noppe, A., Roberts, A. P., Yap, A. S., Gomez, G. A., & Neufeld, Z. (2015). Modelling wound closure in an epithelial cell sheet using the cellular potts model. *Integrative Biology*, *7*(10), 1253–1264.

Radhakrishnan, B., & Sarma, G. (2008). Coupled finite element–potts model simulations of grain growth in copper interconnects. *MRS Online Proceedings Library*, *1156*(1), 1–6.

Rani[1], Y., & Rohil, H. (2013). A study of hierarchical clustering algorithm. *ter S & on Te SIT*, *2*, 113.

Redinz, J. A., & Martins, M. L. (2001). Roughness exponent in two-dimensional percolation, potts model, and clock model. *Physical Review E*, *63*(6), 066133.

Reichardt, J., & Bornholdt, S. (2004). Detecting fuzzy community structures in complex networks with a potts model. *Physical Review Letters*, *93*(21), 218701.

Rodriguez, M. Z., Comin, C. H., Casanova, D., Bruno, O. M., Amancio, D. R., Costa, L. d. F., & Rodrigues, F. A. (2019). Clustering algorithms: A comparative approach. *PloS one*, *14*(1), e0210236.

Roozbeh, M., & Arashi, M. (2016). Shrinkage ridge regression in partial linear models. *Communications in Statistics-Theory and Methods*, *45*(20), 6022–6044.

Rose, K., Gurewitz, E., & Fox, G. (1991). Deterministic annealing, constrained clustering, and optimization.

Ross, G. (1969). Algorithm as 15: single linkage cluster analysis. *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, *18*(1), 106–110.

Ross, G. (1985). Single linkage cluster analysis. *Applied Statistics Algorithms*, (pp. 56–64).

Rutkevich, S. B. (2017). Radiative corrections to the quark masses in the ferromagnetic ising and potts field theories. *Nuclear Physics B*, *923*, 508–557.

Sætrom, J., & Omre, H. (2011). Ensemble kalman filtering with shrinkage regression techniques. *Computational Geosciences*, *15*(2), 271–292.

Salas, J., & Sokal, A. D. (1997). Dynamic critical behavior of the swendsen-wang algorithm: The two-dimensional three-state potts model revisited. *Journal of statistical physics*, *87*(1-2), 1–36.

Sasirekha, K., & Baby, P. (2013). Agglomerative hierarchical clustering algorithm-a. *International Journal of Scientific and Research Publications*, *83*, 83.

Schreiber, N., Cohen, R., & Haber, S. (2018). Ferromagnetic potts models with multisite interaction. *Physical Review E*, *97*(3), 032106.

Scianna, M., & Preziosi, L. (2014). A cellular potts model for the mmp-dependent and-independent cancer cell migration in matrix microtracks of different dimensions. *Computational Mechanics*, *53*(3), 485–497.

Scullard, C. R., & Jacobsen, J. L. (2016). Potts-model critical manifolds revisited. *Journal of Physics A: Mathematical and Theoretical*, *49*(12), 125003.

Seifoddini, H. (1988). Machine grouping—expert systems: Comparison between single linkage and average linkage clustering techniques in forming machine cells. *Computers & industrial engineering*, *15*(1-4), 210–216.

Seifoddini, H. K. (1989). Single linkage versus average linkage clustering in machine cells formation applications. *Computers & Industrial Engineering*, *16*(3), 419–426.

Selke, W., & Huse, D. A. (1983). Interfacial adsorption in planar potts models. *Zeitschrift für Physik B Condensed Matter*, *50*(2), 113–116.

Seo, J., & Shneiderman, B. (2003). Interactively exploring hierarchical clustering results. In *The Craft of Information Visualization*, (pp. 334–340). Elsevier.

Shimagaki, K., & Weigt, M. (2019). Selection of sequence motifs and generative hopfield-potts models for protein families. *Physical Review E*, *100*(3), 032128.

Similä, T., & Tikka, J. (2007). Input selection and shrinkage in multiresponse linear regression. *Computational Statistics & Data Analysis*, *52*(1), 406–422.

Singh, S. P. (2020). The ising model: Brief introduction and its application. In *Solid State Physics-Metastable, Spintronics Materials and Mechanics of Deformable Bodies-Recent Progress*. IntechOpen.

Smith, D., & Smith, M. (2006). Estimation of binary markov random fields using markov chain monte carlo. *Journal of Computational and Graphical Statistics*, *15*(1), 207–227.

Sokal, A. (1997). Monte carlo methods in statistical mechanics: foundations and new algorithms. In *Functional integration*, (pp. 131–192). Springer.

Sprenger, T. C., Brunella, R., & Gross, M. H. (2000). *H-blob: a hierarchical visual clustering method using implicit surfaces*. IEEE.

Stam, A. (1983). Generation of a random partition of a finite set by an urn model. *Journal of Combinatorial Theory, Series A*, *35*(2), 231–240.

Stauffer, D., Hehl, F. W., Ito, N., Winkelmann, V., & Zabolitzky, J. G. (1993). Ising model. In *Computer Simulation and Computer Algebra*, (pp. 79–84). Springer.

Steyerberg, E., Eijkemans, M., & Habbema, J. (2001). Application of shrinkage techniques in logistic regression analysis: a case study. *Statistica Neerlandica*, *55*(1), 76–88.

Storath, M., & Weinmann, A. (2014). Fast partitioning of vector-valued images. *SIAM Journal on Imaging Sciences*, *7*(3), 1826–1852.

Strauss, T., & von Maltitz, M. J. (2017). Generalising ward's method for use with manhattan distances. *PloS one*, *12*(1), e0168288.

Sun, H., Huang, J., Han, J., Deng, H., Zhao, P., & Feng, B. (2010). gskeletonclu: Density-based network clustering via structure-connected tree division or agglomeration. In *2010 IEEE International Conference on Data Mining*, (pp. 481–490). IEEE.

Sundberg, R. (2006). Shrinkage regression. *Encyclopedia of environmetrics*, *5*.

Swat, M. H., Hester, S. D., Heiland, R. W., Zaitlen, B. L., Glazier, J. A., & Shirinifard, A. (????). Compucell3d manual and tutorial version 3.5. 0. *Detail*, *7*(7), 7.

Sweeny, M. (1983). Monte carlo study of weighted percolation clusters relevant to the potts models. *Physical Review B*, *27*(7), 4445.

Swendsen, R. H., & Wang, J.-S. (1987). Nonuniversal critical dynamics in monte carlo simulations. *Physical review letters*, *58*(2), 86.

Szekely, G. J., Rizzo, M. L., et al. (2005). Hierarchical clustering via joint between-within distances: Extending ward's minimum variance method. *Journal of classification*, *22*(2), 151–184.

Szorkovszky, A., Kotrschal, A., Herbert-Read, J. E., Buechel, S. D., Romenskyy, M., Rosén, E., van der Bijl, W., Pelckmans, K., Kolm, N., & Sumpter, D. J. (2018). Assortative interactions revealed by sorting of animal groups. *Animal Behaviour*, *142*, 165–179.

Tan, D., Li, C., Zhu, W., & Jiang, F. (2020). A comprehensive neural networks study of the phase transitions of potts model. *New Journal of Physics*, *22*(6), 063016.

Tan, P.-N., Steinbach, M., & Kumar, V. (2013). Data mining cluster analysis: basic concepts and algorithms. *Introduction to data mining*, (pp. 487–533).

Tanaka, K., Kataoka, S., Yasuda, M., Waizumi, Y., & Hsu, C.-T. (2014). Bayesian image segmentations by potts prior and loopy belief propagation. *Journal of the Physical Society of Japan*, *83*(12), 124002.

Thompson, J. R. (1968). Some shrinkage techniques for estimating the mean. *Journal of the American Statistical Association*, *63*(321), 113–122.

Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society: Series B (Methodological)*, *58*(1), 267–288.

Tomita, Y. (2018). Measurement of entanglement entropy in the two-dimensional potts model using wavelet analysis. *Physical Review E*, *97*(5), 052128.

Tomita, Y., & Okabe, Y. (2001). Probability-changing cluster algorithm for potts models. *Physical Review Letters*, *86*(4), 572.

Tong, D. (2006). Lectures on quantum field theory. *Part III Cambridge University Mathematics Tripos, Michaelmas*.

Tong, D. (2012). Statistical physics. *Part II Cambridge University Mathematics Tripos, Michaelmas*.

Tsoumakas, G., Spyromitros-Xioufis, E., Vilcek, J., & Vlahavas, I. (2020). Datasets from mulan: A java library for multi-label learning.
URL http://mulan.sourceforge.net/datasets-mtr.html

Tsuchiya, K., Nishiyama, T., & Tsujita, K. (2001). A deterministic annealing algorithm for a combinatorial optimization problem using replicator equations. *Physica D: Nonlinear Phenomena*, *149*(3), 161–173.

Tutz, G., & Leitenstorfer, F. (2006). Response shrinkage estimators in binary regression. *Computational statistics & data analysis*, *50*(10), 2878–2901.

Tzortzis, G., & Likas, A. (2014). The minmax k-means clustering algorithm. *Pattern Recognition*, *47*(7), 2505–2516.

Van Erp, S., Oberski, D. L., & Mulder, J. (2019). Shrinkage priors for bayesian penalized regression. *Journal of Mathematical Psychology*, *89*, 31–50.

van Houwelingen, H. C., & Sauerbrei, W. (2013). Cross-validation, shrinkage and variable selection in linear regression revisited.

Vathy-Fogarassy, A., Feil, B., & Abonyi, J. (2005). Minimal spanning tree based fuzzy clustering. *Proc World Acad Sci Eng Tech*, *8*, 7–12.

Vlachos, A., Korhonen, A., & Ghahramani, Z. (2009). Unsupervised and constrained dirichlet process mixture models for verb clustering. In *Proceedings of the workshop on geometrical models of natural language semantics*, (pp. 74–82).

Voss-Böhme, A. (2012). Multi-scale modeling in morphogenesis: a critical analysis of the cellular potts model. *PloS one*, *7*(9), e42852.

Wang, J., Wang, J., & Stanley, H. E. (2018). Multiscale multifractal dcca and complexity behaviors of return intervals for potts price model. *Physica A: Statistical Mechanics and its Applications*, *492*, 889–902.

Wang, J.-S., & Swendsen, R. H. (1990). Cluster monte carlo algorithms. *Physica A: Statistical Mechanics and its Applications*, *167*(3), 565–579.

WANG, X.-l., LIU, Q.-b., LU, C.-h., & HOU, D.-f. (2009). Minimum spanning tree clustering algorithm [j]. *Journal of Chinese Computer Systems*, *5*.

Wang, Y., Liu, X., Dou, Y., & Li, R. (2017). Multiple kernel clustering framework with improved kernels. *Discover*, *1*(2), 3–4.

Wattanachon, U., & Lursinsap, C. (2004). Agglomerative hierarchical clustering for nonlinear data analysis. In *2004 IEEE International Conference on Systems, Man and Cybernetics (IEEE Cat. No. 04CH37583)*, vol. 2, (pp. 1420–1425). IEEE.

Wen, U.-P., Lan, K.-M., & Shih, H.-S. (2009). A review of hopfield neural networks for solving mathematical programming problems. *European Journal of Operational Research*, *198*(3), 675–687.

Wersing, H., & Ritter, H. (1999). Backtracking deterministic annealing for constraint satisfaction problems.

Wilkin, G. A., & Huang, X. (2007). K-means clustering algorithms: implementation and comparison. In *Second International Multi-Symposiums on Computer and Computational Sciences (IMSCCS 2007)*, (pp. 133–136). IEEE.

Wu, F.-Y. (1982). The potts model. *Reviews of modern physics*, *54*(1), 235.

Wu, H.-M., Tien, Y.-J., & Chen, C.-h. (2010). Gap: A graphical environment for matrix visualization and cluster analysis. *Computational Statistics & Data Analysis*, *54*(3), 767–778.

Xing, Y., & Wang, J. (2018). Nonlinear complexity behaviors of agent-based 3d potts financial dynamics with random environments. *Physica A: Statistical Mechanics and its Applications*, *492*, 824–836.

Xiong, S., & Joseph, V. R. (2013). Regression with outlier shrinkage. *Journal of Statistical Planning and Inference*, *143*(11), 1988–2001.

Xu, K., & Wang, J. (2017). Weighted fractional permutation entropy and fractional sample entropy for nonlinear potts financial dynamics. *Physics Letters A*, *381*(8), 767–779.

Xu, N., Finkelman, R. B., Dai, S., Xu, C., & Peng, M. (2021). Average linkage hierarchical clustering algorithm for determining the relationships between elements in coal. *ACS omega*, *6*(9), 6206–6217.

Xu, R., & Wunsch, D. (2008). *Clustering*, vol. 10. John Wiley & Sons.

Yamaguchi, C. (2015). A curious relationship between potts glass models. *The European Physical Journal B*, *88*(8), 1–9.

Yan, B., & Sarkar, P. (2016). On robustness of kernel clustering. *arXiv preprint arXiv:1606.01869*.

Yüzbaşı, B., Arashi, M., & Ejaz Ahmed, S. (2020). Shrinkage estimation strategies in generalised ridge regression models: Low/high-dimension regime. *International Statistical Review*, *88*(1), 229–251.

Zanella, G., et al. (2015). Random partition models and complementary clustering of anglo-saxon place-names. *The Annals of Applied Statistics*, *9*(4), 1792–1822.

Zareamoghaddam, H., Ahmed, S. E., & Provost, S. B. (2020). Shrinkage estimation applied to a semi-nonparametric regression model. *The International Journal of Biostatistics*, *1*(ahead-of-print).

Zhang, L., Zhou, W., & Jiao, L. (2002). Kernel clustering algorithm. *CHINESE JOURNAL OF COMPUTERS-CHINESE EDITION-*, *25*(6), 587–590.

Zhao, R., Ding, C., & Deng, Y. (2018). Overlap of two topological phases in the antiferromagnetic potts model. *Physical Review E*, *97*(5), 052131.

Zhao, X., & Fu, L. (2019). Machine learning phase transition: An iterative proposal. *Annals of Physics*, *410*, 167938.

Zheng, Z., Fan, Y., & Lv, J. (2014). High dimensional thresholded regression and shrinkage effect. *Journal of the Royal Statistical Society: Series B: Statistical Methodology*, (pp. 627–649).

Zhou, M., Favaro, S., & Walker, S. G. (2017). Frequency of frequencies distributions and size-dependent exchangeable random partitions. *Journal of the American Statistical Association*, *112*(520), 1623–1635.

Zou, H., & Hastie, T. (2003). Regression shrinkage and selection via the elastic net, with applications to microarrays. *JR Stat Soc Ser B*, *67*, 301–20.

Main Latex files (.tex)

Click here to access/download
Electronic Supplementary Material
main.tex